



# **Universidad Carlos III de Madrid**

Escuela Politécnica Superior  
Ingeniería en Informática

Proyecto de fin de carrera

## **IMPLEMENTACIÓN DE UN ALGORITMO DE APRENDIZAJE POR REFUERZO EN UNA ARQUITECTURA MULTIAGENTE COOPERATIVA**

Autor

Adrián Antón Collado

Directores

Javier Carbó Rubiera

Fernando Fernández Rebollo



## Resumen

La creciente importancia del fenómeno de la interacción entre sistemas informáticos ha modificado la forma tradicional de analizarlos y programarlos. La necesidad de interacción entre los componentes del sistema es cada vez más importante para poder realizar o resolver tareas conjuntas que de forma individual serían muy costosas o incluso imposibles de desarrollar.

Los **sistemas multiagente**[1] ofrecen una plataforma muy interesante y completa para la realización de estas tareas, pero además ofrecen la posibilidad de incluir en cada uno de los agentes un comportamiento dotado de una inteligencia que puede evolucionar gracias a técnicas de **aprendizaje automático**[2][3].

Las técnicas de **aprendizaje por refuerzo**[19][20][21], encuadradas dentro del aprendizaje automático, son muy adecuadas para su uso junto con agentes debido a la integración con el entorno necesaria que implementa el agente, de la que aprenden las técnicas de aprendizaje por refuerzo, y a la ejecución de acciones sobre el entorno obtenidas por las técnicas de aprendizaje por refuerzo y llevadas a cabo por el agente.

Sin embargo, el aprendizaje por refuerzo en sistemas multiagente[22] ofrece nuevos retos derivados de la **distribución del aprendizaje**, como pueden ser la necesidad de la **coordinación** entre agentes o la **distribución del conocimiento**, que deben ser analizados y tratados.

En el presente proyecto se aborda el problema del aprendizaje por refuerzo en sistemas multiagente siguiendo la **metodología Gaia**[6][7] e implementándolo sobre la **plataforma de agentes JADE**[13]. El problema tratado es el de, dado un estado inicial, aprender una política que conduzca al estado final. Los agentes se situarán en un entorno en el que otros agentes pueden tener el mismo objetivo o un objetivo diferente que les haga enfrentarse. Además usarán estructuras de gestión del conocimiento para su puesta en común.

El objetivo de este proyecto es resolver el clásico problema de aprendizaje en el que un agente tiene que aprender a **llegar a una meta dentro de un entorno desconocido**. Será modificado de forma que serán varios los agentes presentes en el entorno. Cada agente tendrá asociada una meta, pudiendo ser compartida con otros agentes o no. El hecho de que la meta sea compartida permite que los agentes compartan el conocimiento adquirido para ayudarse mutuamente en la obtención de una solución. La meta puede ser incluso opuesta, lo que implica no sólo que los agentes no puedan compartir el conocimiento, sino que pueda existir una cierta competencia entre ellos.



# Índice general

<b>RESUMEN.....</b>	<b>3</b>
<b>1. MOTIVACIÓN Y OBJETIVOS.....</b>	<b>13</b>
1.1. MOTIVACIÓN .....	13
1.2. OBJETIVOS .....	13
1.3. VISIÓN GENERAL DEL DOCUMENTO .....	14
<b>2. ESTADO DEL ARTE .....</b>	<b>17</b>
2.1. AGENTES SOFTWARE.....	17
2.2. SISTEMAS MULTIAGENTE .....	19
2.3. ARQUITECTURAS PARA SISTEMAS MULTIAGENTE .....	22
2.4. METODOLOGÍAS DE DESARROLLO PARA SISTEMAS MULTIAGENTE .....	28
2.5. PLATAFORMAS .....	36
2.6. APRENDIZAJE POR REFUERZO .....	39
2.7. Q-LEARNING.....	44
2.8. APRENDIZAJE POR REFUERZO EN SISTEMAS MULTIAGENTE .....	49
2.9. CONCLUSIONES Y VISIÓN GENERAL DEL SISTEMA.....	53
<b>3. ANÁLISIS DEL SISTEMA.....</b>	<b>57</b>
3.1. DESCRIPCIÓN DEL PROCESO DE APRENDIZAJE .....	57
3.2. ORGANIZACIONES.....	61
3.3. MODELO DEL ENTORNO.....	63
3.4. MODELO PRELIMINAR DE ROLES .....	70
3.5. MODELO PRELIMINAR DE INTERACCIONES.....	71
3.6. NORMAS DE LA ORGANIZACIÓN.....	72
<b>4. DISEÑO ARQUITECTÓNICO .....</b>	<b>82</b>
4.1. REPRESENTACIÓN DE LA ESTRUCTURA ORGANIZATIVA .....	82
4.2. INTEGRACIÓN DEL ALGORITMO DE APRENDIZAJE EN LA ARQUITECTURA .....	86
4.3. MODELO DE ROLES .....	86
4.4. MODELO DE INTERACCIONES .....	88
<b>5. DISEÑO DETALLADO .....</b>	<b>89</b>
5.1. DEFINICIÓN DE LOS PROCESOS DE APRENDIZAJE.....	89
5.2. DEFINICIÓN DEL MODELO DE AGENTES .....	93
5.3. DEFINICIÓN DEL MODELO DE SERVICIOS .....	98
5.4. DEFINICIÓN DE LOS PROTOCOLOS .....	99
<b>6. EXPERIMENTOS.....</b>	<b>117</b>
6.1. MÉTODO DE EVALUACIÓN .....	117
6.2. ENTORNOS DE PRUEBA .....	119
6.3. CONFIGURACIÓN DE LOS EXPERIMENTOS .....	120
6.4. EXPERIMENTOS EN EL ENTORNO PUERTA.....	121
6.5. EXPERIMENTOS EN EL ENTORNO PASILLO .....	132
6.6. CONCLUSIONES DE LA EXPERIMENTACIÓN .....	144
<b>7. CONCLUSIONES Y TRABAJOS FUTUROS .....</b>	<b>147</b>
7.1. CONCLUSIONES.....	147
7.2. TRABAJOS FUTUROS .....	149
<b>8. REFERENCIAS .....</b>	<b>153</b>
<b>ANEXO I: MODELO DE ROLES.....</b>	<b>155</b>
I.1. MODELO PREELIMINAR DE ROLES .....	155

I.2. MODELO COMPLETO DE ROLES.....	159
<b>ANEXO II: MODELO DE INTERACCIONES .....</b>	<b>169</b>
II.1. MODELO PRELIMINAR DE INTERACCIONES.....	169
II.2. MODELO COMPLETO DE INTERACCIONES .....	172
<b>ANEXO III: COMPORTAMIENTOS ASOCIADOS A LOS AGENTES .....</b>	<b>179</b>
III.1. COMPORTAMIENTOS ASOCIADOS AL AGENTE ESTADO .....	179
III.2. COMPORTAMIENTOS ASOCIADOS AL AGENTE MOVIMIENTO.....	179
III.3. COMPORTAMIENTOS ASOCIADOS AL AGENTE REINICIO .....	180
III.4. COMPORTAMIENTOS ASOCIADOS AL AGENTE GESTOR ESTADÍSTICO.....	180
III.5. COMPORTAMIENTOS ASOCIADOS AL AGENTE PROCESADOR ESTADÍSTICO.....	181
III.6. COMPORTAMIENTOS ASOCIADOS AL AGENTE GESTOR DE CONOCIMIENTO EXCLUSIVO .....	182
III.7. COMPORTAMIENTOS ASOCIADOS AL AGENTE GESTOR DE CONOCIMIENTO COMPARTIDO ....	182
III.8. COMPORTAMIENTOS ASOCIADOS AL AGENTE PROCESADOR DE CONOCIMIENTO.....	183
III.9. COMPORTAMIENTOS ASOCIADOS AL AGENTE JUGADOR EN ENTORNO EXCLUSIVO .....	183
III.10. COMPORTAMIENTOS ASOCIADOS AL AGENTE JUGADOR EN ENTORNO COMPARTIDO ....	184
III.11. COMPORTAMIENTOS ASOCIADOS AL AGENTE CONTROL DEL ENTRENAMIENTO EN ENTORNO EXCLUSIVO .....	185
III.12. COMPORTAMIENTOS ASOCIADOS AL AGENTE CONTROL DEL ENTRENAMIENTO EN ENTORNO COMPARTIDO .....	185
<b>ANEXO IV: PARÁMETROS CONFIGURABLES DE LOS AGENTES.....</b>	<b>187</b>
IV.1. PARÁMETROS CONFIGURABLES DEL AGENTE ESTADO .....	187
IV.2. PARÁMETROS CONFIGURABLES DEL AGENTE MOVIMIENTO .....	187
IV.3. PARÁMETROS CONFIGURABLES DEL AGENTE REINICIO .....	187
IV.4. PARÁMETROS CONFIGURABLES DEL AGENTE GESTOR ESTADÍSTICO.....	187
IV.5. PARÁMETROS CONFIGURABLES DEL AGENTE PROCESADOR ESTADÍSTICO.....	188
IV.6. PARÁMETROS CONFIGURABLES DEL AGENTE GESTOR DE CONOCIMIENTO EXCLUSIVO .....	188
IV.7. PARÁMETROS CONFIGURABLES DEL AGENTE GESTOR DE CONOCIMIENTO COMPARTIDO ....	188
IV.8. PARÁMETROS CONFIGURABLES DEL AGENTE JUGADOR EN ENTORNO EXCLUSIVO .....	188
IV.8. PARÁMETROS CONFIGURABLES DEL AGENTE JUGADOR EN ENTORNO COMPARTIDO .....	189
<b>ANEXO V: MODELO DE SERVICIOS .....</b>	<b>191</b>
<b>ANEXO VI: MENSAJES DEL SISTEMA.....</b>	<b>195</b>
<b>ANEXO VII: ONTOLOGÍA .....</b>	<b>207</b>
VII.1. CONCEPTOS .....	207
VII.2. ACCIONES .....	210
VII.3. PREDICADOS .....	214
<b>ANEXO VIII: NOTACIÓN DE LAS NORMAS DE LA ORGANIZACIÓN .....</b>	<b>219</b>
<b>ANEXO IX: GESTIÓN Y PLANIFICACIÓN DEL PROYECTO.....</b>	<b>221</b>
IX.1 RECURSOS MATERIALES .....	221
IX.2 PLANIFICACIÓN .....	221
IX.3 RECURSOS HUMANOS .....	222
IX.4 TOTALES.....	223

# Índice de ilustraciones

Ilustración 1 : Agente y su relación con el entorno .....	17
Ilustración 2 : Sistema homogéneo y sistema heterogéneo .....	19
Ilustración 3 : Sistema multiagente de organización estática.....	20
Ilustración 4 : Sistema multiagente de organización dinámica.....	20
Ilustración 5 : Sistema multiagente con control distribuido y sistema multiagente con control central.....	20
Ilustración 6 : Sistema multiagente no comunicativo y sistema multiagente comunicativo .....	21
Ilustración 7 : Arquitectura BDI .....	24
Ilustración 8 : Arquitectura de subsunción .....	26
Ilustración 9 : Arquitectura TouringMachines .....	27
Ilustración 10 : Arquitectura FIPA .....	28
Ilustración 11 : Metodología INGENIAS y sus diferentes aspectos.....	32
Ilustración 12 : Fases de la metodología Gaia .....	34
Ilustración 13 : Fases de la metodología Gaia II .....	36
Ilustración 14 : Remote Agent Management GUI de JADE .....	38
Ilustración 15 : Agente Sniffer de JADE.....	38
Ilustración 16 : Modelo de aprendizaje por refuerzo.....	40
Ilustración 17 : Ejemplo de proceso de decisión de Markov .....	42
Ilustración 18 : Algoritmo de Q-learning.....	45
Ilustración 19 : Aproximación con redes de neuronas de la función $Q(s,a)$ [21] .....	49
Ilustración 20 : Ejemplo de obstáculos en un entorno .....	54
Ilustración 21 : Casillas de inicio y casillas meta en un entorno.....	54
Ilustración 22 : Ejemplo de entorno de dos equipos.....	55
Ilustración 23 : Algoritmo de Q-learning inicial .....	57
Ilustración 24 : Algoritmo de Q-learning adaptado para establecer las comunicaciones necesarias .....	58
Ilustración 25 : Algoritmo de Q-learning adaptado para establecer las comunicaciones necesarias e interactuar con el agente de control en cada ciclo .....	58
Ilustración 26 : Ejemplo de interbloqueo entre agentes.....	59
Ilustración 27 : Algoritmo de Q-learning adaptado para establecer las comunicaciones necesarias e interactuar con el agente de control en cada ciclo y en cada episodio .....	59
Ilustración 28 : Ejemplo de $\epsilon$ – <b>greedy</b> con $\epsilon$ variable, sólo incrementos .....	61
Ilustración 29 : Ejemplo de $\epsilon$ – <b>greedy</b> con $\epsilon$ variable, con incrementos y decrementos. ....	61
Ilustración 30 : Ejemplo de entorno Tablero.....	64
Ilustración 31 : Política obtenida mediante Q-Learning .....	64
Ilustración 32 : Acciones asociadas al entorno Repositorio.....	65
Ilustración 33 : Acciones asociadas al entorno Tablero.....	67
Ilustración 34 : Ejemplo de entorno Tablero.....	67
Ilustración 35 : Acciones asociadas al entorno Estadístico .....	70
Ilustración 36 : Organizaciones presentes en el sistema.....	82
Ilustración 37 : Algoritmo de decisión de la estrategia de movimiento a seguir.....	91
Ilustración 38 : Ejemplo de $\epsilon$ – <b>greedy</b> variable.....	92
Ilustración 39 : Algoritmo de retardo del proceso de aprendizaje .....	93
Ilustración 40 : Agentes de la organización Tablero y sus roles .....	94
Ilustración 41 : Agentes de la organización Estadística y sus roles .....	95
Ilustración 42 : Agentes de la organización Repositorio y sus roles.....	96
Ilustración 43 : Agente Jugador en Entorno Exclusivo y sus roles .....	96
Ilustración 44 : Agente Jugador en Entorno Compartido y sus roles.....	97
Ilustración 45 : Agentes de la organización Control y sus roles .....	98
Ilustración 46 : Ejemplo de gráfico para dos jugadores .....	117
Ilustración 47 : Ejemplo de gráfico para cuatro jugadores .....	118
Ilustración 48 : Ejemplo de comparativa entre dos procesos de aprendizaje .....	118
Ilustración 49 : Entorno puerta.....	119
Ilustración 50 : Entorno pasillo.....	119
Ilustración 51 : Comparativa las distintas configuraciones de agentes para el caso base en el entorno puerta .....	121
Ilustración 52: Comparativa entre el experimento 1 y el caso base .....	123

Ilustración 53 : Comparativa entre el experimento 2 y el caso base .....	125
Ilustración 54: Comparativa entre el experimento 3 y el caso base .....	127
Ilustración 55: Comparativa entre el experimento 4 y el caso base .....	129
Ilustración 56 : Comparativa entre el experimento 5 y el caso base .....	131
Ilustración 57 : Comparativa las distintas configuraciones de agentes para el caso base en el entorno pasillo .....	132
Ilustración 58 : Comparativa entre el experimento 6 y el caso base .....	135
Ilustración 59 : Comparativa entre el experimento 7 y el caso base .....	137
Ilustración 60: Comparativa entre el experimento 8 y el caso base .....	139
Ilustración 61: Comparativa entre el experimento 9 y el caso base .....	141
Ilustración 62 : Comparativa entre el experimento 10 y el caso base .....	143
Ilustración 63: Planificación del proyecto en diagramas de Gantt.....	222
Protocolo 1 : Pedir Estado.....	99
Protocolo 2 : Pedir Reinicio .....	100
Protocolo 3 : Pedir Movimiento.....	101
Protocolo 4 : Enviar Estadísticas Entorno Exclusivo .....	102
Protocolo 5 : Enviar Estadísticas Entorno Compartido .....	103
Protocolo 6 : Actualizar grafica .....	104
Protocolo 7 : Enviar política exclusiva.....	104
Protocolo 8 : Enviar política compartida.....	105
Protocolo 9 : Pedir mejor movimiento en entorno exclusivo .....	106
Protocolo 10 : Pedir política compartida .....	106
Protocolo 11 : Empezar entrenamiento exclusivo .....	107
Protocolo 12 : Empezar episodio de entrenamiento compartido.....	107
Protocolo 13 : Empezar ciclo de entrenamiento compartido.....	108
Protocolo 14 : Listo para entrenamiento exclusivo .....	108
Protocolo 15 : Entrenamiento exclusivo acabado.....	109
Protocolo 16 : Informar entrenamiento en entorno exclusivo acabado .....	109
Protocolo 17 : Listo para episodio de entrenamiento compartido .....	110
Protocolo 18 : Acabado episodio de entrenamiento compartido .....	110
Protocolo 19 : Listo para ciclo de entrenamiento compartido .....	111
Protocolo 20 : Acabado ciclo de entrenamiento en entorno compartido .....	111
Protocolo 21 : Entrenamiento en entorno compartido acabado .....	112
Protocolo 22 : Pedir registro en entorno Tablero.....	112
Protocolo 23 : Solicitud de Uso.....	113
Protocolo 24 : Pedir registro .....	114
Protocolo 25 : Encontrar agente .....	114
Protocolo 26 : Finalizar Ejecución.....	115



## Índice de tablas

Tabla 1 : Función de transición del proceso de decisión de Markov .....	42
Tabla 2 : Tabla Q óptima del ejemplo .....	48
Tabla 3 : Información estática para el entorno Tablero de la ilustración 27.....	68
Tabla 4 : Información dinámica para el entorno Tablero de la ilustración 27.....	68
Tabla 5: Modelo de servicios.....	99
Tabla 6 : Configuración para la ejecución del caso base en el entorno puerta .....	121
Tabla 7 : Configuración del experimento 1.....	122
Tabla 8 : Configuración del experimento 2.....	124
Tabla 9 : Configuración del experimento 3.....	126
Tabla 10 : Configuración del experimento 4.....	128
Tabla 11 : Configuración del experimento 5.....	130
Tabla 12 : Configuración para la ejecución del caso base en el entorno pasillo.....	132
Tabla 13 : Configuración del experimento 6.....	134
Tabla 14 : Configuración del experimento 7.....	136
Tabla 15 : Configuración del experimento 8.....	138
Tabla 16 : Configuración del experimento 9.....	140
Tabla 17 : Configuración del experimento 10.....	142
Tabla 18 : Rol preliminar Gestor de Estado .....	155
Tabla 19 : Rol preliminar Gestor de Reinicio.....	155
Tabla 20 : Rol preliminar Gestor de Movimiento .....	155
Tabla 21 : Rol preliminar Gestor Estadístico.....	156
Tabla 22 : Rol preliminar Procesador Estadístico.....	156
Tabla 23 : Rol preliminar Gestor de Conocimiento en Entorno Exclusivo .....	156
Tabla 24 : Rol preliminar Gestor de Conocimiento en Entorno Compartido .....	157
Tabla 25 : Rol preliminar Procesador de Conocimiento .....	157
Tabla 26 : Rol preliminar Jugador en Entorno Exclusivo .....	157
Tabla 27 : Rol preliminar Jugador en Entorno Compartido .....	158
Tabla 28 : Rol preliminar Control del Entrenamiento en Entorno Exclusivo .....	158
Tabla 29 : Rol preliminar Control del Entrenamiento en Entorno Compartido .....	159
Tabla 30 : Rol Gestor de Estado .....	159
Tabla 31 : Rol Gestor de Reinicio .....	160
Tabla 32 : Rol Gestor de Movimiento.....	160
Tabla 33 : Rol Gestor Estadístico .....	161
Tabla 34 : Rol Procesador Estadístico.....	161
Tabla 35 : Rol Gestor de Conocimiento en Entorno Exclusivo .....	162
Tabla 36 : Rol Gestor de Conocimiento en Entorno Compartido .....	162
Tabla 37 : Rol Procesador de Conocimiento.....	163
Tabla 38 : Rol Jugador en Entorno Exclusivo .....	163
Tabla 39 : Rol Jugador en Entorno Compartido .....	164
Tabla 40 : Rol Control del Entrenamiento en Entorno Exclusivo.....	165
Tabla 41 : Rol Control del Entrenamiento en Entorno Compartido .....	166
Tabla 42 : Rol Registrador DF .....	166
Tabla 43 : Rol Buscador de Agentes .....	167
Tabla 44 : Rol Servidor.....	167
Tabla 45 : Rol Procesador.....	167
Tabla 46 : Rol Finalizable.....	167
Tabla 47 : Interacción preliminar Pedir Estado.....	169
Tabla 48 : Interacción preliminar Pedir Reinicio .....	169
Tabla 49 : Interacción preliminar Pedir Movimiento.....	169
Tabla 50 : Interacción preliminar Enviar Estadísticas .....	169
Tabla 51 : Interacción preliminar Enviar Conocimiento .....	169
Tabla 52 : Interacción preliminar Pedir Conocimiento Exclusivo.....	170
Tabla 53 : Interacción preliminar Pedir Política Compartida .....	170
Tabla 54 : Interacción preliminar Empezar Entrenamiento Exclusivo .....	170
Tabla 55 : Interacción preliminar Empezar Episodio de Entrenamiento Compartido .....	170
Tabla 56 : Interacción preliminar Empezar Ciclo de Entrenamiento Compartido.....	170
Tabla 57 : Interacción preliminar Listo para Entrenamiento Exclusivo .....	171

Tabla 58 : Interacción preliminar Entrenamiento Exclusivo Acabado.....	171
Tabla 59 : Interacción preliminar Listo para Episodio de Entrenamiento Compartido.....	171
Tabla 60 : Interacción preliminar Acabado Episodio de Entrenamiento Compartido .....	171
Tabla 61 : Interacción preliminar Listo para Ciclo de Entrenamiento Compartido .....	171
Tabla 62 : Interacción preliminar Acabado Ciclo de Entrenamiento Colectivo.....	172
Tabla 63 : Interacción preliminar Entrenamiento Colectivo Acabado .....	172
Tabla 64 : Interacción preliminar Pedir Registro en Entorno Tablero.....	172
Tabla 65 : Interacción Pedir Estado.....	172
Tabla 66 : Interacción Pedir Reinicio .....	172
Tabla 67 : Interacción Pedir Movimiento.....	173
Tabla 68 : Interacción Enviar Estadísticas Entorno Exclusivo .....	173
Tabla 69 : Interacción Enviar Estadísticas Entorno Compartido .....	173
Tabla 70 : Interacción Actualizar Grafica .....	173
Tabla 71 : Interacción Enviar Política Exclusiva.....	173
Tabla 72 : Interacción Enviar Política Compartida.....	174
Tabla 73 : Interacción Pedir Mejor Movimiento en Entorno Exclusivo.....	174
Tabla 74 : Interacción Pedir Política Compartida .....	174
Tabla 75 : Interacción Empezar Entrenamiento Exclusivo.....	174
Tabla 76 : Interacción Empezar Episodio de Entrenamiento Compartido .....	174
Tabla 77 : Interacción Empezar Ciclo de Entrenamiento Compartido .....	175
Tabla 78 : Interacción Listo para Entrenamiento Exclusivo .....	175
Tabla 79 : Interacción Entrenamiento Exclusivo Acabado.....	175
Tabla 80 : Interacción Listo para Episodio de Entrenamiento Compartido.....	175
Tabla 81 : Interacción Acabado Episodio de Entrenamiento Compartido .....	175
Tabla 82 : Interacción Listo para Ciclo de Entrenamiento Compartido .....	176
Tabla 83 : Interacción Acabado Ciclo de Entrenamiento en Entorno Compartido .....	176
Tabla 84 : Interacción Entrenamiento en Entorno Compartido Acabado.....	176
Tabla 85 : Interacción Pedir Registro en Entorno Tablero .....	176
Tabla 86 : Interacción Informar Entrenamiento en Entorno Exclusivo Acabado .....	176
Tabla 87 : Interacción Solicitud de Uso.....	177
Tabla 88 : Interacción Pedir Registro.....	177
Tabla 89 : Interacción Encontrar Agente.....	177
Tabla 90 : Interacción Finalizar Ejecución.....	177
Tabla 91 : Comportamientos asociados al agente estado .....	179
Tabla 92 : Comportamientos asociados al agente movimiento.....	180
Tabla 93 : Comportamientos asociados al agente reinicio .....	180
Tabla 94 : Comportamientos asociados al agente gestor estadístico .....	181
Tabla 95 : Comportamientos asociados al agente procesador estadístico.....	181
Tabla 96 : Comportamientos asociados al agente gestor conocimiento exclusivo .....	182
Tabla 97 : Comportamientos asociados al agente gestor conocimiento compartido .....	183
Tabla 98 : Comportamientos asociados al agente procesador de conocimiento.....	183
Tabla 99 : Comportamientos asociados al agente jugador en entorno exclusivo .....	184
Tabla 100 : Comportamientos asociados al agente jugador en entorno compartido .....	185
Tabla 101 : Comportamientos asociados al agente control del entrenamiento en entorno exclusivo.....	185
Tabla 102 : Comportamientos asociados al agente control del entrenamiento en entorno compartido.....	186
Tabla 103 : Parámetros configurables del agente estado.....	187
Tabla 104 : Parámetros configurables del agente movimiento .....	187
Tabla 105 : Parámetros configurables del agente reinicio .....	187
Tabla 106 : Parámetros configurables del agente gestor estadístico.....	187
Tabla 107 : Parámetros configurables del agente procesador estadístico .....	188
Tabla 108 : Parámetros configurables del agente gestor de conocimiento exclusivo.....	188
Tabla 109 : Parámetros configurables del agente gestor de conocimiento compartido.....	188
Tabla 110 : Parámetros configurables del agente Jugador en Entorno Exclusivo.....	188
Tabla 111 : Parámetros configurables del agente Jugador en Entorno Compartido.....	190
Tabla 112 : Servicio Obtener Estado .....	191
Tabla 113 : Servicio Ejecutar Movimiento .....	191
Tabla 114 : Servicio Pedir Reinicio .....	191
Tabla 115 : Servicio Enviar estadísticas entorno Exclusivo.....	192

Tabla 116 : Servicio Enviar estadísticas entorno Compartido.....	192
Tabla 117 : Servicio Actualizar gráfica.....	192
Tabla 118 : Servicio Recibir política exclusiva .....	193
Tabla 119 : Servicio Recibir política compartida .....	193
Tabla 120 : Servicio Obtener mejor movimiento en entorno exclusivo.....	193
Tabla 121 : Servicio Obtener política en entorno compartida .....	194
Tabla 122 : Servicio Solicitud de uso .....	194
Tabla 123 : Servicio Pedir registro.....	194
Tabla 124 : Servicio Finalizar ejecución.....	194
Tabla 125: Recursos materiales empleados .....	221
Tabla 126: Planificación temporal del proyecto y sus costes.....	221
Mensaje 1 : Pedir Estado .....	195
Mensaje 2 : Estado Exclusivo Agente.....	195
Mensaje 3 : Estado Compartido Agente .....	195
Mensaje 4 : Estado Exclusivo Desconocido.....	196
Mensaje 5 : Estado Compartido Desconocido .....	196
Mensaje 6 : Pedir Reinicio.....	196
Mensaje 7 : Reinicio Incorrecto .....	196
Mensaje 8 : Pedir Movimiento .....	197
Mensaje 9 : Movimiento Correcto .....	197
Mensaje 10 : Movimiento Incorrecto .....	197
Mensaje 11 : Enviar Estadísticas Exclusivas.....	198
Mensaje 12 : Enviar Estadísticas Compartidas .....	198
Mensaje 13 : Actualizar Gráfica.....	198
Mensaje 14 : Enviar Política Exclusiva .....	199
Mensaje 15 : Enviar Política Compartida .....	199
Mensaje 16 : Pedir Mejor Movimiento Exclusivo .....	199
Mensaje 17 : Pedir Política Compartida.....	200
Mensaje 18 : Listo Para Entrenamiento En Entorno Exclusivo .....	200
Mensaje 19 : Empezar Entrenamiento Entorno Exclusivo .....	200
Mensaje 20 : Entrenamiento Entorno Exclusivo Acabado .....	200
Mensaje 21 : Listo Para Episodio Entrenamiento En Entorno Compartido .....	201
Mensaje 22 : Empezar Episodio Entrenamiento Entorno Compartido .....	201
Mensaje 23 : Acabado Episodio Entrenamiento En Entorno Compartido .....	201
Mensaje 24 : Listo Para Ciclo Entrenamiento En Entorno Compartido .....	201
Mensaje 25 : Empezar Ciclo Entrenamiento En Entorno Compartido .....	202
Mensaje 26 : Acabado Ciclo Entrenamiento En Entorno Compartido .....	202
Mensaje 27 : Entrenamiento En Entorno Compartido Acabado.....	202
Mensaje 28 : Pedir Registro En Entorno.....	202
Mensaje 29 : Agente Registrado .....	203
Mensaje 30 : Agente No Registrado .....	203
Mensaje 31 : Solicitar Uso Agente.....	203
Mensaje 32 : Agente Autorizado.....	203
Mensaje 33 : Agente No Autorizado .....	204
Mensaje 34 : Finalizar Ejecución .....	204
Mensaje 35 : Ejecucion No Finalizada.....	204
Mensaje 36 : Mejor Movimiento Exclusivo .....	205
Mensaje 37 : Mejor Movimiento Compartido .....	205
Concepto 1 : Agente .....	207
Concepto 2 : Estado Exclusivo .....	207
Concepto 3 : Equipo.....	207
Concepto 4 : Estado Compartido.....	208
Concepto 5 : Estadísticas Ciclo Entorno Compartido .....	208
Concepto 6 : Estadísticas Ciclo Entorno Exclusivo .....	209
Concepto 7 : Motivo .....	209
Concepto 8 : Movimiento.....	209
Concepto 9 : Refuerzo .....	209
Concepto 10 : Valor Q.....	210

Concepto 11 : Tupla Q Exclusiva .....	210
Concepto 12 : Tupla Q Compartida .....	210
Acción 1 : Empezar Entrenamiento Entorno Exclusivo.....	210
Acción 2 : Enviar Estadísticas Exclusivas .....	211
Acción 3 : Finalizar Ejecución.....	211
Acción 4 : Entrenamiento Exclusivo Acabado .....	211
Acción 5 : Listo Para Entrenamiento Entorno Exclusivo .....	211
Acción 6 : Pedir Estado .....	211
Acción 7 : Pedir Movimiento .....	212
Acción 8 : Pedir Registro En Tablero.....	212
Acción 9 : Reiniciar .....	212
Acción 10 : Solicitar Uso Agente .....	212
Acción 11 : Listo Para Ciclo Entrenamiento En Entorno Compartido.....	212
Acción 12 : Empezar Ciclo Entrenamiento En Entorno Compartido.....	213
Acción 13 : Acabado Ciclo Entrenamiento En Entorno Compartido.....	213
Acción 14 : Listo Para Episodio Entrenamiento En Entorno Compartido .....	213
Acción 15 : Empezar Episodio Entrenamiento En Entorno Compartido .....	213
Acción 16 : Acabado Episodio Entrenamiento En Entorno Compartido .....	214
Acción 17 : Enviar Estadísticas Compartidas.....	214
Acción 18 : Entrenamiento Compartido Acabado.....	214
Predicado 1 : Agente Autorizado .....	214
Predicado 2 : Agente No Autorizado.....	215
Predicado 3 : Agente Registrado.....	215
Predicado 4 : Agente No Registrado .....	215
Predicado 5 : Agente Pertenece Equipo .....	215
Predicado 6 : Ejecucion Finalizada.....	215
Predicado 7 : Ejecucion No Finalizada .....	216
Predicado 8 : Estado Agente Compartido .....	216
Predicado 9 : Estado Agente Exclusivo .....	216
Predicado 10 : Estado Desconocido .....	216
Predicado 11 : Inicio.....	217
Predicado 12 : Mejor Movimiento .....	217
Predicado 13 : Meta.....	217
Predicado 14 : Movimiento Correcto.....	217
Predicado 15 : Movimiento Incorrecto.....	218
Predicado 16 : Reinicio Incorrecto.....	218

# 1. Motivación y objetivos

Un agente software se define como un sistema informático autónomo para la realización de sus objetivos, capaz de reaccionar a los estímulos del entorno que le rodea e incluso capaz de desarrollar un comportamiento proactivo, modificando el entorno que le rodea para lograr su objetivo. Dicho comportamiento puede ser adquirido usando técnicas de aprendizaje automático, de forma que el comportamiento puede evolucionar a través de la experiencia.

Los sistemas multiagente son aquellos en los que varios agentes conviven en la realización de sus tareas. Como resultado de esa convivencia pueden coordinarse, cooperar e incluso competir. Para que convivan es necesario el desarrollo de un lenguaje de comunicación, con el cual los agentes pueden compartir conocimiento y experiencias para ayudarse mutuamente en sus procesos de aprendizaje.

## 1.1. Motivación

Las propiedades emergentes de los sistemas multiagente hacen muy interesante afrontar problemas de inteligencia artificial distribuida, puesto que ofrecen nuevos recursos y técnicas que explorar. Además, características como poder compartir la experiencia y el conocimiento adquirido dan un valor añadido a dichas técnicas que es necesario estudiar.

Pero los sistemas multiagente llevan asociados problemas que no tienen los sistemas monoagente, como el hecho de gestionar una comunicación antes inexistente o negociaciones en entornos compartidos que pueden implicar ceder eficiencia en un agente para mejorar la eficiencia global del sistema.

## 1.2. Objetivos

El objetivo principal de este proyecto es la implementación de una arquitectura multiagente en la que la colaboración entre agentes consiga resolver problemas aprendiendo de la experiencia obtenida mediante técnicas de aprendizaje por refuerzo.

Más concretamente, los objetivos de este proyecto son:

- Explorar y conocer las nuevas metodologías de creación de sistemas multiagente desde un punto de vista práctico. Particularmente, la metodología para la creación de agentes Gaia.

- Explorar y conocer las plataformas de creación de sistemas multiagente también desde un punto de vista práctico. En concreto, conocer a fondo las características de la plataforma de creación de agentes JADE.
- Construir un lenguaje, compuesto por una ontología común y una serie de protocolos, para la comunicación.
- Construir un sistema multiagente en el que los agentes consigan mayor rendimiento por el hecho de estar en sociedad que estando aislados.
- Implementar un sistema distribuido que utilice aprendizaje por refuerzo, en el cual los agentes puedan compartir su experiencia con otros agentes con los que comparten objetivos e integrar dicha experiencia en el proceso de aprendizaje por refuerzo.
- Probar las diferentes opciones que ofrece el sistema e incorporar técnicas que refinan el proceso de aprendizaje.

Para ello, se construirá un sistema en el que varios agentes parten de un estado inicial y tienen que alcanzar un estado final. El conjunto de estados será común para todos los agentes pero su uso será exclusivo. Es decir, cualquier agente podrá encontrarse en cualquier estado siempre y cuando sea el único que se encuentra en dicho estado.

Los agentes presentes en el sistema compartirán información mediante los mensajes de la ontología compuesta y los protocolos diseñados para intentar conseguir mayor rendimiento en sus tareas.

### **1.3. Visión general del documento**

Este documento sigue la metodología de desarrollo Gaia para la creación de un sistema multiagente, ampliándola en aquellas partes en las que es necesario desarrollar ciertos temas no contemplados en la metodología.

En el capítulo “*Estado del arte*” se resumen las bases teóricas del trabajo realizado. Se abordan arquitecturas para sistemas multiagente y metodologías para su desarrollo, así como las bases de las técnicas de aprendizaje por refuerzo y, por último, se detallan algunas de esas técnicas aplicadas a sistemas multiagente. Además se expone el sistema que se quiere construir, detallando la forma del entorno en el que ejecutarán los agentes y cuál será su objetivo dentro del sistema.

En el capítulo “*Análisis del sistema*” se detalla la fase de Análisis según Gaia, con los modelos preliminares de roles e interacciones, así como las normas que la organización de agentes debe cumplir y hacer cumplir. Esta parte se amplía con la detallada descripción del algoritmo de aprendizaje adoptado y sus técnicas asociadas.

En el capítulo “*Diseño arquitectónico*” se cumple dicha fase según Gaia, precisando la forma de las organizaciones del sistema multiagente, así como el modelo de roles e interacciones completado, mostrando cómo se integra el proceso de aprendizaje con la arquitectura propuesta.

En el capítulo “*Diseño detallado*” se exponen, según Gaia, tanto el modelo de agentes del sistema como el modelo de servicios. Además, se añade la descripción de todos los protocolos mediante diagramas de secuencia.

En el capítulo “*Pruebas*” se incluye la colección de pruebas del sistema y de sus distintas características. Además, analizando el rendimiento de dichas características y la mejora que aportan a los distintos procesos de aprendizaje.

En el capítulo “*Conclusiones y trabajos futuros*” se analizan el logro de los objetivos propuestos para el presente proyecto y se proponen mejoras y aplicaciones para las técnicas empleadas.

En el capítulo “*Anexos*” se detallan partes del desarrollo del proyecto, como el modelo de roles o el modelo de interacciones o el conjunto completo de pruebas realizadas sobre el sistema implementado.





## 2. Estado del arte

### 2.1. Agentes Software

Los agentes software pueden definirse como programas con un comportamiento humanoide. Pueden funcionar en los equipos de los usuarios (PDA's, PC's, teléfonos móviles, sistemas GPS, etc) y en los nodos de comunicación de las redes. Tienen autonomía y capacidad de decisión, siendo capaces de razonar y aprender. Pueden comunicarse, organizarse y desplazarse colaborando o compitiendo con otros agentes. En general, usan sus capacidades para ayudar al usuario a resolver sus problemas de forma inteligente y proactiva.

No existe una definición precisa de “Agente Software” en términos computacionales. Una de las definiciones más citadas [Wooldridge, 1997] es la siguiente: “Un agente es un sistema informático situado en un entorno y que es capaz de realizar acciones de forma autónoma para conseguir sus objetivos de diseño”. Pero esta definición, aunque identifica alguna de las características esenciales de un agente, como la autonomía, no permite distinguirlo claramente de un sistema distribuido convencional.

En general, un agente software es un sistema informático situado en algún entorno dentro del cual actúa de forma autónoma y flexible para así cumplir sus objetivos. De esta forma, el agente se encuentra en un entorno del que recibe, a través sus sensores, información que utilizará para ejecutar, mediante sus actuadores, las acciones que considere para alcanzar su objetivo.

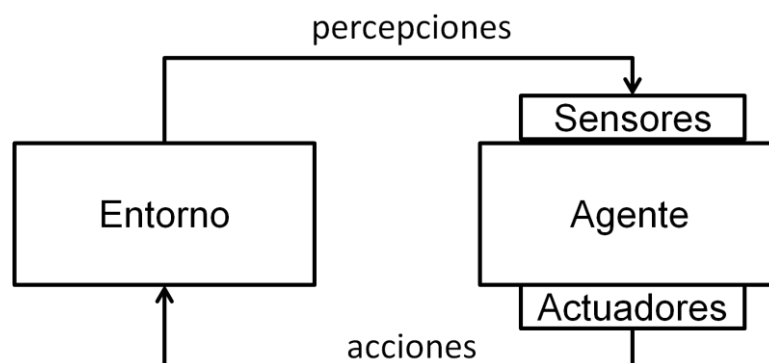


Ilustración 1 : Agente y su relación con el entorno

Aunque es difícil definir el término “Agente Software”, se puede dar una lista de características generales comunes a todos los agentes.

### **2.1.1. Características de un agente**

Las características principales de un agente son:

- **Autonomía.** No necesita la directa intervención de otros sistemas informáticos o humanos en sus mecanismos de decisión. Tiene absoluta capacidad de decisión sobre sus acciones.
- **Sociabilidad.** Capacidad para comunicarse con los elementos que le rodean, ya sea el entorno u otros agentes presentes en el mismo, para negociar, cooperar o competir, usando algún lenguaje o método de comunicación.
- **Reactividad.** Capacidad de responder a los estímulos del ambiente, recibidos en sus sensores, con las acciones adecuadas y en el tiempo adecuado.
- **Proactividad.** Capacidad de tomar la iniciativa, sin necesidad de ningún estímulo externo, en la realización de las acciones necesarias para cumplir con los objetivos para los que es diseñado.

Además un agente puede tener otras características como:

- **Movilidad.** Capacidad de moverse en el entorno que le rodea e incluso desplazarse entre entornos.
- **Adaptabilidad.** Capacidad para trabajar en diferentes redes, plataformas y sistemas operativos de software. Además, capacidad para enfrentarse a los problemas y resolverlos sin la intervención del usuario.
- **Robustez.** Capacidad de solucionar las tareas que le han sido asignadas en cualquier entorno.
- **Aprendizaje.** Capacidad de aprender del entorno y de las acciones que ejecuta en él.
- **Racionalidad.** La racionalidad de un agente depende de:
  - La medida de rendimiento que define el criterio de éxito.
  - El conocimiento del medio en que habita acumulado por el agente.
  - Las acciones que el agente puede llevar a cabo.
  - La secuencia de percepciones, entendiendo por tal todo aquello que, hasta ese momento, haya percibido el agente.

## 2.2. Sistemas multiagente

La parte más importante de un agente es la propiedad emergente que se obtiene cuando se acompaña de otros agentes. La interacción autónoma entre ellos genera un valor agregado con respecto a otro agente inteligente.

### 2.2.1. Tipos de sistemas multiagente

Existen diferentes sistemas multiagente atendiendo a la homogeneidad de sus agentes, su organización, su comportamiento y su comunicación.

Un sistema multiagente puede ser **homogéneo** o **heterogéneo**, dependiendo de la estructura de sus agentes. La heterogeneidad agrega robustez al sistema, pero su construcción es más costosa y compleja.

Por ejemplo, si intentamos implementar un equipo de fútbol mediante un sistema multiagente, como en la ilustración 2, con un sistema multiagente homogéneo podemos construir un equipo en el que todos los jugadores tengan las mismas funciones (todos intentan buscar el balón y llevarlo hasta la portería contraria). Sin embargo, con un sistema multiagente heterogéneo podemos asignar a cada jugador una tarea dependiendo del rol que debe asumir.

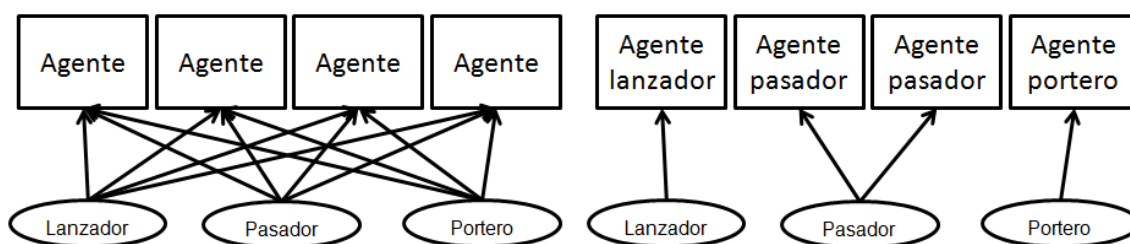


Ilustración 2 : Sistema homogéneo y sistema heterogéneo

La organización de un sistema multiagente puede estar programada a priori o no; es decir, puede ser **estática** o **dinámica**. En una estructura estática como la que se muestra en la ilustración 3, los agentes responsables de una tarea determinada son siempre los mismos, mientras que en una estructura dinámica como la que se muestra en la ilustración 4, los agentes pueden crearse y destruirse y sus tareas pueden cambiar. Por ejemplo, retomando el ejemplo del equipo de fútbol, en función del éxito o fracaso de sus miembros en las diferentes tareas, unos agentes pueden encargarse de un conjunto de tareas según vayan aprendiendo a hacerlas, dejando otras tareas para que otros agentes se especialicen en ellas.

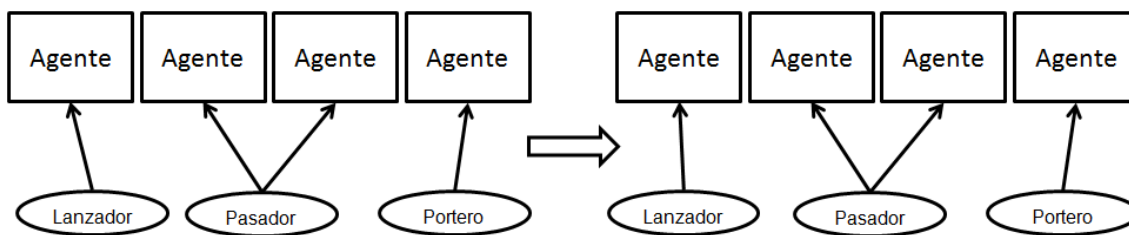


Ilustración 3 : Sistema multiagente de organización estática

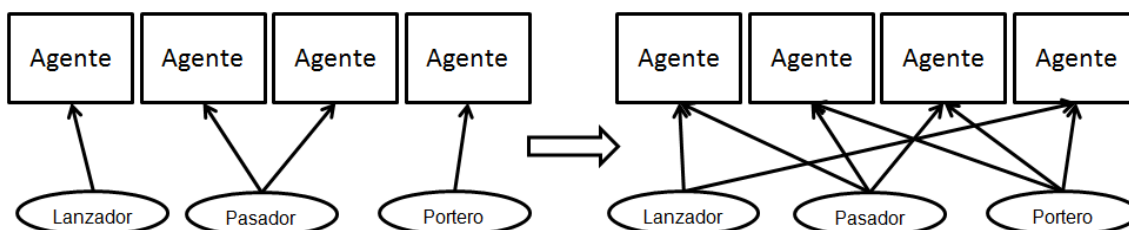


Ilustración 4 : Sistema multiagente de organización dinámica

Desde el punto de vista del comportamiento en un sistema multiagente se puede optar por dos alternativas. En la primera el comportamiento es estructurado de forma jerárquica, con un **control central** sobre el comportamiento general del sistema. En la segunda alternativa, el **control es distribuido** entre todos los agentes y la combinación de sus comportamientos, obteniendo un comportamiento de control emergente del sistema.

En el equipo de fútbol multiagente de la ilustración 5, un nuevo agente control puede dirigir las tareas de todos los agentes jugadores de una forma ordenada o bien dicho control ordenado puede emerger de los mismos jugadores.

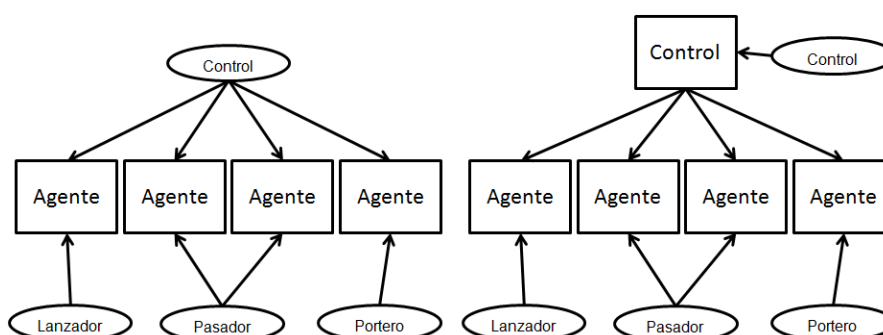
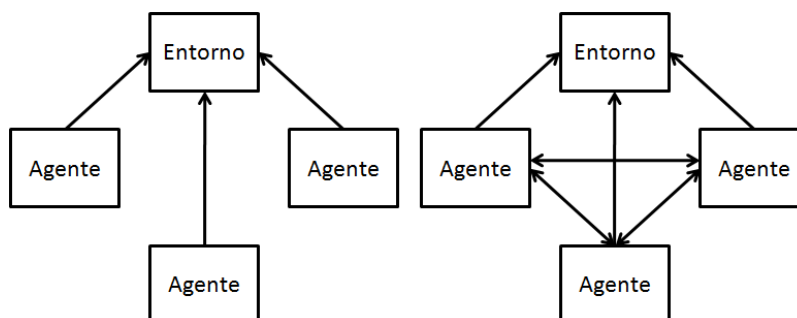


Ilustración 5 : Sistema multiagente con control distribuido y sistema multiagente con control central

En cuanto a la comunicación entre los agentes pertenecientes a un sistema multiagente, existen nuevamente dos alternativas. En primer lugar, el sistema multiagente puede ser un sistema **no comunicativo** en el que los agentes sólo tienen en cuenta sus sensores para conocer el estado de los otros agentes. O bien, un sistema multiagente

**comunicativo**, pudiéndose hablar de una interacción real, mediante protocolos o lenguajes definidos que permiten un conocimiento más preciso del estado del sistema, comportamientos cooperativos, de coordinación y de negociación.

En el equipo de fútbol multiagente de la ilustración 6 los jugadores pueden comunicarse para lograr su objetivo cooperando entre ellos activamente, siendo un sistema comunicativo, o los agentes pueden alcanzar su objetivo sin comunicación, intentando lograr su objetivo por sí mismos, siendo un sistema no comunicativo.



**Ilustración 6 : Sistema multiagente no comunicativo y sistema multiagente comunicativo**

### **2.2.2. Propiedades de los sistemas multiagente**

Los sistemas multiagente tienen una serie de propiedades relativas a la descripción de las competencias de sus integrantes, al modelo de agentes conocidos, al modelo de comunicación entre agentes, al comportamiento global de los agentes y a los puntos de interacción de los agentes.

La descripción de las competencias de los agentes define el problema a resolver en términos de tareas, subtareas y relaciones entre ellas, determinando cómo resolver el problema, cómo distribuirlo entre los distintos agentes y las interacciones de los mismos. La descomposición del problema en tareas se hace en dos fases:

- La primera durante la definición del problema, en la fase de diseño del sistema multiagente.
- La segunda durante la resolución del problema, asignando dinámicamente la actividad de los agentes.

El modelo de agentes conocidos es la información que cada agente conoce de otros agentes. Dicha información puede contener:

- Capacidades de los agentes externos; es decir, las tareas que pueden cumplir.

- Conocimientos de los agentes externos; es decir, aquella información que otros agentes han recopilado.
- Creencias de los agentes externos; es decir, aquella información acerca de otros agentes que no se tiene por segura.

De esta forma, el modelo de agentes se usa para intentar predecir el comportamiento de otros agentes, planear y coordinar actividades locales de acuerdo a una meta global y para permitir una estructura flexible en la arquitectura del agente.

El modelo de comunicación entre agentes define los canales de comunicación entre ellos, permitiendo que uno tenga conocimiento sobre los otros agentes, y protocolos y lenguajes, que permiten que la comunicación se lleve a cabo.

El comportamiento global coherente de los agentes en un sistema debe ser siempre el primer objetivo de dicho sistema, entendiendo por *comportamiento global coherente* aquel que lleva a una solución global medida por su eficiencia, calidad y claridad.

### **2.3. Arquitecturas para sistemas multiagente**

La arquitectura define los mecanismos que permiten interconectar los componentes que hacen que un sistema multiagente se comporte como tal. Especifica cómo se descomponen los agentes en un conjunto de organizaciones que interactúan entre sí para lograr la funcionalidad requerida.

Existen tres tipos: las arquitecturas deliberativas, las arquitecturas reactivas y las arquitecturas híbridas, que se basan en la visión de los sistemas multiagente como arquitecturas de sistemas intencionales.

#### **2.3.1. Sistemas intencionales**

Se denomina *sistema intencional* a aquella entidad cuyo comportamiento se puede predecir por el método de atribuirles creencias, deseos y perspicacia racional. Si el sistema es muy sencillo, atribuirle descripciones intencionales consistentes con el comportamiento observado no resulta de gran utilidad. Pero en sistemas complejos las nociones intencionales son herramientas de abstracción que proporcionan una forma adecuada y familiar de describir, explicar y predecir el comportamiento de estos sistemas.

Un agente puede verse como un sistema capaz de definirse correctamente mediante posturas intencionales, puesto que contiene actitudes que proporcionan información y pro-actitudes. Las de información están relacionadas con el conocimiento que un agente tiene

sobre su entorno, mientras que las pro-actitudes son aquellas que guían de algún modo las acciones del agente.

No hay un consenso claro acerca de cuál es la combinación entre las actitudes de información y las pro-actitudes más adecuada para definir un agente. Lo razonable parece que dicha combinación posea, al menos, una actitud de información y una pro-actitud. Entre los modelos más aceptados se encuentran:

- **Conocimiento y acción:** Se centra en definir qué datos debería conocer un agente para así proyectar el conjunto de precondiciones en el de acciones. De esta manera un agente podría ejecutar las acciones en un orden determinado como consecuencia de un estímulo.
- **Intención:** Se utilizan dos actitudes básicas: creencias y objetivos. Las actitudes, como la intención, se definen en términos de las creencias.
- **Creencias, deseos e intenciones:** Da lugar a la arquitectura BDI (*Belief, Desire, Intention*), en la que se profundizará más adelante.
- **Intenciones, creencias, conocimiento y habilidades:** Desarrolla una familia de lógicas para representar intenciones, creencias, conocimiento, habilidades y comunicación utilizando una extensión del modelo de tiempo, caracterizado por incorporar un formalismo rico y complejo.

### 2.3.2. Arquitecturas deliberativas

Arquitecturas deliberativas son las que se basan en la teoría clásica de planificación, es decir,

- Parten de un estado inicial en el cual empieza la resolución del problema.
- Generan un conjunto de planes.
- Alcanzan un estado meta.

Las arquitecturas deliberativas usan modelos de representación simbólica del conocimiento, por lo que una arquitectura deliberativa contiene un modelo simbólico del mundo, explícitamente representado, en el que las decisiones se toman utilizando mecanismos de razonamiento lógico, basados en la manipulación simbólica, con el propósito de alcanzar los objetivos del agente.

Es por ello que a la hora de usar una arquitectura deliberativa es esencial una descripción simbólica adecuada del entorno del problema e integrarla en el sistema multiagente para que los agentes puedan razonar y llevar a cabo sus tareas encomendadas.

### 2.3.3. Arquitectura BDI

La arquitectura BDI (*Belief, Desire, Intention*) es una arquitectura deliberativa que se caracteriza porque los agentes que la implementan están dotados de creencias, deseos e intenciones.

La vista interna realiza el modelado de cada clase del agente a través de tres modelos:

- El **modelo de creencias** representa el conocimiento que el agente tiene del entorno, es decir, la forma de representar el estado del entorno. Son las variables o las expresiones simbólicas mediante las cuales el agente representa el estado del entorno.
- El **modelo de deseos** contiene los objetivos del agente. Un deseo puede ser hacer que una variable alcance un valor o una expresión simbólica. Un deseo representa un estado final, por lo que el objetivo de todo agente es alcanzarlo.
- Para alcanzar los deseos propuestos a partir de las creencias existentes es necesario definir un mecanismo de planificación que nos permita identificar las intenciones, un plan que debe ser ejecutado y reconsiderado cuando no ofrezca los resultados esperados. Estas intenciones, vinculadas a la consecución de un objetivo, constituyen el **modelo de planificación** del agente.

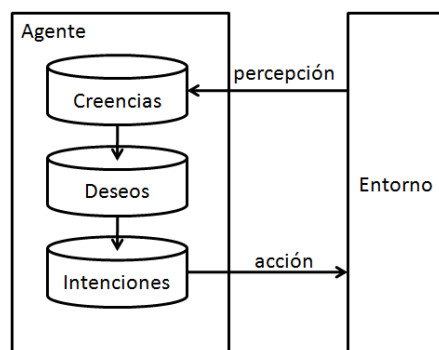


Ilustración 7 : Arquitectura BDI

La arquitectura BDI divide su desarrollo en dos vistas, la **vista interna** y la **vista externa**. El desarrollo de la **vista interna** comienza con el análisis de los diferentes planes para alcanzar un objetivo. Los planes para responder a un evento o alcanzar un objetivo se describen empleando una gráfica similar a los diagramas de estados, extendiendo la notación para expresar fallos en un plan.

La **vista externa** consiste en la descomposición del sistema en agentes y la definición de sus interacciones. Se realiza empleando dos modelos,



- El **modelo de agente**, que describe las relaciones jerárquicas entre clases de agentes y las relaciones entre agentes concretos, y
- El **modelo de interacción**, que describe las responsabilidades, servicios e interacciones entre los agentes y los sistemas externos.

El proceso de desarrollo de la vista externa comienza con la identificación de los papeles (funcionales, organizativos, etc.) para identificar los agentes y organizarlos en una jerarquía de clases de agentes (*modelo de clases de agentes*). A continuación se identifican las responsabilidades asociadas a cada papel y los servicios proporcionados y empleados para cumplir con estas responsabilidades e interacciones realizadas en los servicios.

Para cada servicio se identifican las interacciones necesarias, los actos comunicativos de dichas interacciones y el contenido de información de las mismas. Toda esta información se recoge en el modelo de ejemplares de agentes.

#### **2.3.4. Arquitecturas reactivas**

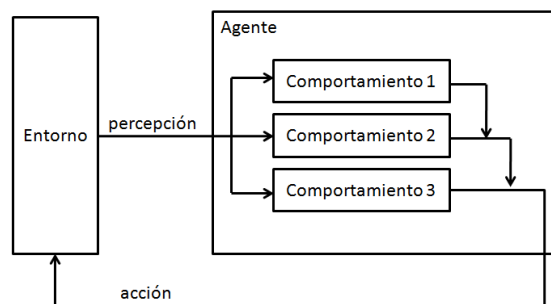
Las arquitecturas reactivas se caracterizan por reaccionar rápidamente a los estímulos externos, careciendo del elemento central de razonamiento de un modelo simbólico y por no utilizar razonamientos simbólicos complejos.

La necesidad de actuar en un entorno impredecible y altamente cambiante dificulta la adopción de una arquitectura deliberativa, ya que las necesidades de replanificación y de continua adaptación del plan a la realidad hacen muy difícil que una arquitectura deliberativa responda con suficiente agilidad.

#### **2.3.5. Arquitectura de subsunción**

La arquitectura de subsunción es una arquitectura reactiva según la cual la inteligencia es una propiedad emergente de ciertos sistemas complejos que permite generar comportamientos inteligentes sin necesidad de construir un modelo simbólico.

Las arquitecturas de subsunción, como la que podemos observar en la ilustración 8, manejan jerarquías de tareas que definen un comportamiento y suelen estar organizadas por capas, de mayor a menor nivel de abstracción. De esta forma las decisiones de una capa pueden inhibir las decisiones de las capas inferiores, haciendo que comportamientos inmediatos frente a estímulos tengan preferencia sobre comportamientos cuya elaboración requiere un mayor tiempo de razonamiento.



**Ilustración 8 : Arquitectura de subsunción**

### **2.3.6. Arquitecturas híbridas**

Tanto las arquitecturas deliberativas como las arquitecturas reactivas presentan ciertas limitaciones. Por ello, aparecen arquitecturas híbridas que combinan aspectos de ambos modelos. Una primera propuesta puede ser construir un agente compuesto de dos subsistemas;

- Un primer sistema deliberativo, que utilice un modelo simbólico y que genere planes.
- Un segundo sistema reactivo, centrado en reaccionar ante los eventos que tengan lugar en el entorno de forma rápida.

Las arquitecturas híbridas suelen estar estructuradas por capas, dado lugar a dos tipos de estructuración según el acceso de cada capa

- Arquitecturas híbridas verticales, donde sólo una capa tiene acceso a los sensores y a los actuadores.
- Arquitecturas híbridas horizontales, donde todas las capas tienen acceso a los sensores y a los actuadores.

La mayoría de las arquitecturas híbridas se estructuran en tres capas:

- Capa reactiva. En ella se toman decisiones acerca de lo que hacer en base a los estímulos recibidos del entorno en tiempo real.
- Capa de conocimiento. Se centra en el conocimiento que el agente posee del medio y normalmente cuenta con una representación simbólica del mismo.
- Capa social. En la que se manejan los aspectos sociales del entorno, incluyendo tanto la información de otros agentes como sus propios deseos e intenciones.

### 2.3.7. Arquitectura *TouringMachines*

La arquitectura *TouringMachines* es un tipo de arquitectura híbrida en la que cada nivel está continuamente sugiriendo qué acción realizar. Cuenta con un sistema de control para garantizar el correcto funcionamiento del agente y se estructura en tres capas horizontales:

- Reactiva, para reaccionar inmediatamente a los posibles cambios del entorno.
- Planificadora, que ejecuta comportamientos proactivos basados en planes.
- Modeladora, que modela el mundo para anticipar conflictos.

Como se puede observar en la ilustración 9, las tres capas están embebidas en un subsistema de control basado en reglas que pueden inhibir entradas y salidas y determina qué capa tiene control sobre el agente.

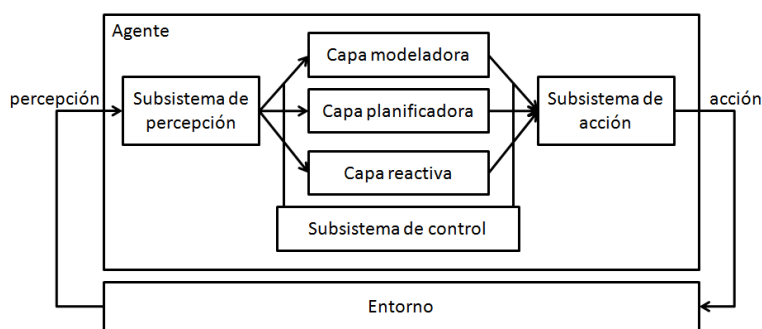


Ilustración 9 : Arquitectura *TouringMachines*

### 2.3.8. Arquitectura *FIPA*

El desarrollo de aplicaciones complejas, compuestas por agentes que interactúan entre sí, necesita del desarrollo de estándares que faciliten estas tareas. FIPA[17] establece los elementos básicos que forman parte de un sistema multiagente. Según este modelo los agentes son procesos software que se encuentran en la plataforma y que interactúan mediante un lenguaje de comunicación de agentes.

La Plataforma de Agentes FIPA proporciona la infraestructura necesaria para poder desarrollar y utilizar agentes. Además, FIPA define los servicios que debe proporcionar la plataforma de agentes: un sistema encargado del transporte de mensajes (**STM**), un sistema de gestión de agentes (**AMS**), un servicio de directorio (**DF**) y un canal de comunicaciones. Cada uno de estos servicios, excepto el de transporte de mensajes, es suministrado por agentes especializados, lo cual supone que la comunicación con ellos se realiza mediante mensajes ACL y la ontología definida para ese servicio.

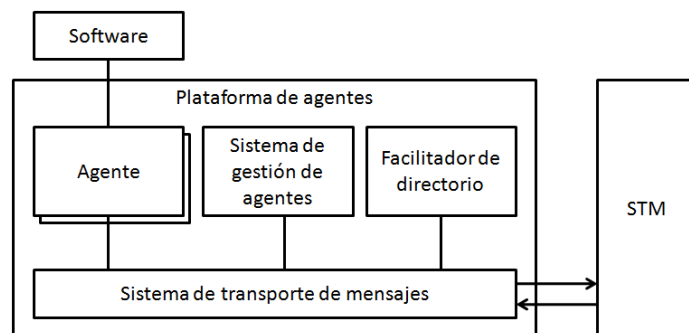


Ilustración 10 : Arquitectura FIPA

- **Sistema de gestión de agentes** (*AMS Agent Management System*). El AMS es el elemento de gestión principal que conoce en todo momento el estado de su plataforma y de los agentes que se están ejecutando en ella. Proporciona no sólo los servicios básicos de creación, destrucción y control de agentes sino que también tiene un servicio de nombres (*Agent Name Service-ANS*), que asocia el nombre a su dirección de transporte real. Otra de las tareas del AMS es el control del ciclo de vida, y los estados de los agentes.
- **Servicio de directorio** (*DF Directory Facilitator*). Como complemento al Servicio de Nombres la plataforma de agentes incluye un servicio de páginas amarillas que permite buscar un agente por sus capacidades. Para poder desempeñar esta tarea los agentes se registran en el DF indicando cuáles son los servicios que ofrecen.
- **Canal de comunicaciones** (*ACC Agent Communication Channel*). Todos los agentes FIPA deben tener acceso a un canal de comunicaciones encargado de gestionar el envío de mensajes. La gestión consiste en el encaminamiento de los mensajes ACL desde su agente origen a su agente destino.
- **Sistema de transporte de mensajes** (*STM*). El STM representa toda la infraestructura de comunicaciones que permite la comunicación entre agentes que pertenecen a la misma plataforma o a plataformas distintas.

## 2.4. Metodologías de desarrollo para sistemas multiagente

La construcción de un sistema multiagente integra tecnologías de distintas áreas de conocimiento:

- Técnicas de ingeniería del software, para estructurar el proceso de desarrollo.
- Técnicas de inteligencia artificial, que capacitan a los programas para tratar situaciones imprevistas y tomar decisiones.
- Técnicas de programación concurrente y distribuida, para tratar la coordinación de tareas ejecutadas en diferentes máquinas bajo diferentes políticas de planificación.

Aunque existen plataformas, de las que hablaremos más tarde, que facilitan el desarrollo de sistemas multiagente, quedan incompletas sin un proceso de desarrollo de software especializado para agentes que haga similar la creación de un sistema multiagente a la producción de software convencional.

Las técnicas convencionales de ingeniería no tienen en cuenta las necesidades de especificación de los sistemas multiagente derivadas de su carácter distribuido: el intercambio de información mediante lenguajes de comunicación orientados a agentes, la movilidad del código o la motivación de los componentes del sistema.

Por ello, se están proponiendo nuevas metodologías basadas en agentes que parten de un modelo de cómo debe ser un sistema multiagente y describen métodos y guías para su construcción.

#### **2.4.1. MAS-CommonKADS**

Esta metodología extiende el modelo definido en CommonKADS[9] agregando técnicas de metodologías orientadas a objetos. La metodología CommonKADS gira en torno al modelo de experiencia y está pensada para desarrollar sistemas expertos que interactúan con el usuario. De hecho considera sólo dos agentes básicos, el usuario y el sistema. Este hecho influye en el modelo de comunicación que únicamente trata de interacciones entre el usuario y el agente.

Esta metodología ha sido la primera en hacer un planteamiento de sistemas multiagente integrados con un ciclo de vida de software, concretamente en espiral, dirigido por riesgos. Propone siete modelos para la definición del sistema:

- Agente
- Tarea
- Experiencia
- Coordinación
- Comunicación
- Organización
- Diseño

Cada modelo reseña la teoría sobre la que se basa. El modelo en sí parte de una descripción gráfica que luego se complementa con explicaciones en lenguaje natural de cada elemento. Por cada modelo existe una descripción de las dependencias respecto de otros modelos y de las actividades involucradas, complementándose con otras notaciones para describir el comportamiento de los agentes cuando interactúan.

La especificación de sistemas multiagente que proporciona MAS-CommonKADS detalla la mayoría de aspectos en lenguaje natural. Esta particularidad dificulta el análisis automático de la especificación generada y supone una gran desventaja frente a semiformalismos como UML, soportado por muchas herramientas.

#### **2.4.2. ZEUS**

ZEUS[8] consta de una herramienta y una metodología. Desde su aparición ZEUS se ha convertido en una referencia de cómo debe ser una herramienta para el desarrollo de sistemas multiagente. Sobre todo por la forma en que combina los distintos resultados de investigación en agentes (planificación, ontologías, asignación de responsabilidades, relaciones sociales entre agentes) en un sistema completamente funcional. De hecho, la aplicación genera incluso programas para arrancar el sistema especificado e incluye herramientas de monitorización.

- Visor de sociedad, que muestra los agentes existentes y sus relaciones.
- Herramienta de control, para ver o modificar remotamente el estado de los agentes.
- Generadores de informes, para obtener estadísticas de funcionamiento e informes de actuación de la sociedad de agentes.

La metodología ZEUS propone un desarrollo en cuatro fases:

- Análisis del dominio
- Diseño de los agentes
- Realización de los agentes
- Soporte en tiempo de ejecución

Las fases soportadas por la herramienta de control son las dos últimas, basándose en el uso de roles para analizar el dominio y en su asignación a agentes.

El ámbito de la metodología se limita a estudiar cómo agrupar la funcionalidad del sistema dentro de cada rol, dejando aparte consideraciones acerca de cómo organizar las tareas, definir las ontologías y las dependencias sociales, aspectos modelables dentro de la herramienta.

#### **2.4.3. MaSE**

MaSE (*Multiagent systems Software Engineering*)[11] se concibe como una abstracción del paradigma orientado a objetos donde los agentes son especializaciones de objetos. En lugar de simples objetos, con métodos que pueden invocarse desde otros objetos, los agentes

se coordinan unos con otros vía conversaciones y actúan proactivamente para alcanzar metas individuales y del sistema.

En MaSE los agentes son simplemente una abstracción conveniente, que puede o no poseer inteligencia. En este sentido, los componentes inteligentes y no inteligentes se gestionan igualmente dentro del mismo armazón. El sistema se construye sobre tecnología orientada a objetos y su aplicación a la especificación y diseño de sistemas multiagente.

La fase de análisis en MaSE consta de tres etapas:

- Capturar los objetivos
- Aplicar los casos de uso.
- Refinar roles.

La fase de diseño consta de cuatro etapas:

- Crear clases de agentes.
- Construir conversaciones.
- Ensamblar clases de agentes.
- Diseñar el sistema.

La mayoría de estos pasos se ejecutan dentro de la herramienta que soporta MaSE, *AgentTool*. Como productos de estas etapas, MaSE espera:

- Diagramas de secuencia para especificar interacciones
- Diagramas de estados para representar procesos internos a las tareas y modelar interacciones
- Descomposición del sistema (agente) en subsistemas (componentes del agente)
- Interconexión de los mismos (definición de la arquitectura del agente).

Pero MaSE es incapaz de mostrar elementos característicos de los sistemas multiagente como pueden ser

- Razonamiento de los agentes
- Organización de los agentes

#### **2.4.4. INGENIAS**

INGENIAS[28] propone el análisis y diseño de sistemas multiagente desde cinco puntos de vista para capturar los diferentes aspectos de un sistema multiagente:

- **Agente**, que describe las responsabilidades con tareas y roles. También considera el control del agente definiendo sus objetivos y los estados que se requieren durante su ejecución.
- **Organización**, que describe el marco en el que existen los agentes, los recursos, las tareas y el propósito del sistema. Para definir la organización hay que considerar su estructura, relaciones sociales y funcionalidad. La estructura determina la arquitectura del sistema descrita en términos de grupos y flujos de trabajo. Los flujos de trabajo relacionan tareas, los recursos asociados a las mismas y sus responsabilidades.
- **Entorno**, que define los sensores y actuadores de los agentes. También identifica los recursos, agentes y aplicaciones existentes con las que tienen que interactuar los agentes.
- **Tareas y objetivos**, cuyo propósito es justificar la ejecución de tareas en función de los objetivos proporcionando la descomposición de tareas y objetivos. Entre ambos hay relaciones especializadas que determinan la información que es necesaria para considerar que un objetivo se ha satisfecho o no. Finalmente, este punto de vista detalla aspectos de bajo nivel de las tareas, como los recursos que necesitan para su ejecución, los módulos software que utilizan, y sus entradas y salidas.
- **Interacción**, que describe cómo se produce la coordinación entre los agentes. La descripción va más allá de lo que serían los diagramas de secuencia o colaboración en UML ya que muestra también la motivación de los participantes en la interacción. Para ello incluye información del estado que requieren los agentes en su ejecución así como las tareas que ejecutarán. Esto permite justificar a nivel de diseño por qué los agentes participan en una interacción y por qué deben continuar.

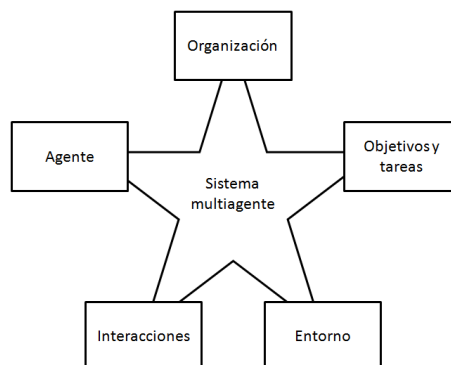


Ilustración 11 : Metodología INGENIAS y sus diferentes aspectos.



#### 2.4.5. *Gaia*

Gaia[6] es una metodología para el diseño de sistemas basados en agentes cuyo objetivo es obtener un sistema que maximice alguna medida de calidad global. Gaia pretende ayudar al analista a ir sistemáticamente desde unos requisitos iniciales a un diseño que, según los autores, esté lo suficientemente detallado como para ser implementado directamente.

En Gaia, el objetivo del análisis consiste en conseguir comprender el sistema y su estructura sin referenciar ningún aspecto de implementación. Esto se consigue a través de la idea de organización. Una organización en Gaia es una colección de roles, los cuales mantienen ciertas relaciones con otros y toman parte en patrones institucionalizados de interacción con otros roles. Los roles agrupan cuatro aspectos:

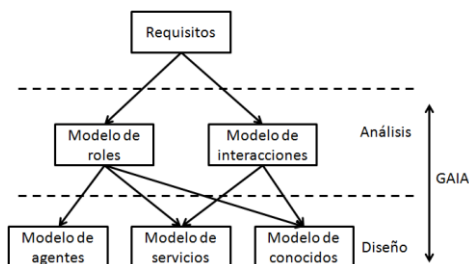
- Responsabilidad del agente
- Recursos que se le permite utilizar
- Tareas asociadas
- Interacciones

Gaia propone trabajar inicialmente con un análisis de alto nivel (fase de análisis en la ilustración 12) en el que se usan dos modelos.

- **Modelo de roles** para identificar los roles clave en el sistema junto con sus propiedades definitorias.
- **Modelo de interacciones**, que define las interacciones mediante una referencia a un modelo institucionalizado de intercambio de mensajes.

Tras esta etapa, se entraría en lo que Gaia considera diseño a alto nivel (fase de diseño en la ilustración 12). El objetivo de este diseño es generar tres modelos:

- **Modelo de agentes**, que define los tipos de agente que existen, cuántas instancias de cada tipo y qué papeles juega cada agente.
- **Modelo de servicios**, que identifica los servicios asociados a cada rol.
- **Modelo de comunicación**, que define los enlaces de comunicación que existen entre los agentes.



**Ilustración 12 : Fases de la metodología Gaia**

A partir de aquí se aplicarían técnicas de diseño orientado a objetos. Sin embargo, esto queda fuera del ámbito de Gaia ya que esta metodología sólo busca especificar cómo una sociedad de agentes colabora para alcanzar los objetivos del sistema y qué se requiere de cada uno para lograrlo.

La principal crítica que se le puede hacer a Gaia es que se queda a un nivel de abstracción demasiado alto. Según los autores, con ello consigue desacoplarse de las distintas soluciones de implementación de agentes.

Otro aspecto discutible es el uso combinado de fórmulas lógicas con fichas de documentación clásicas en ingeniería del software. En Gaia parece asumirse que poniendo fórmulas lógicas se consigue una mejor comprensión del problema y una mayor exactitud. El problema de todas estas fórmulas, aparte de su comprensión, es su definición.

#### **2.4.6. Gaia II**

Basándose en el trabajo realizado para desarrollar la metodología Gaia [6] los autores deciden extender dicha metodología creando Gaia II [7] explotando nuevas abstracciones organizativas y extendiendo significativamente el rango de aplicaciones en el que Gaia puede ser aplicado.

Gaia II (en adelante, Gaia), incluye tres fases en su proceso

- **Fase de análisis** que recoge y organiza la especificación sentando la base del diseño de la organización.
- **Fase de diseño arquitectónico**, donde se define la arquitectura del sistema multiagente.
- **Fase de diseño detallado**, cuyo propósito es proporcionar un diseño del sistema multiagente que pueda ser fácilmente implementado.

La **fase de análisis** en Gaia incluye la identificación de:

- Los **objetivos de las organizaciones** que constituyen el sistema y su comportamiento global, identificando cómo descomponer fructíferamente la organización global en varias suborganizaciones.
- El **modelo del entorno**, entendido como una representación abstracta del entorno en el que el sistema multiagente se situará.
- El **modelo preliminar de roles**, que identifica las habilidades básicas requeridas por la organización. Este modelo contiene sólo aquellos roles, posiblemente no definidos completamente, que deben ser identificados sin comprometer el diseño con una arquitectura concreta.
- El **modelo preliminar de interacciones**, que identifica las interacciones básicas que requieren los roles preliminares para cumplir sus objetivos. De nuevo, las interacciones deben ser identificadas sin comprometer el diseño con una arquitectura concreta.
- Las **normas que la organización** debe respetar y hacer cumplir en su comportamiento global. Dichas reglas deben expresar constantes de la ejecución de actividades de cada rol y protocolos que sean importantes para la ejecución del sistema multiagente.

La salida de la fase de análisis, consistente en un modelo del entorno, un modelo preliminar de roles, un modelo preliminar de interacciones y un conjunto de normas organizativas, es explotada en las fases de diseño arquitectónico y de diseño detallado. La **fase de diseño arquitectónico** incluye:

- **La definición de la estructura organizativa del sistema** en términos de su topología y su régimen de control. Esta actividad debe considerar la eficiencia de la organización, la organización del mundo real (si existe) en el que el sistema está situado y el cumplimiento de las normas organizativas.
- **Completar los modelos preliminares de roles y de interacciones**, basándose en la estructura organizativa adoptada y separando siempre que sea posible los aspectos dependientes de la organización de los independientes.

Una vez que la arquitectura del sistema está identificada junto con sus modelos de roles e interacciones, empieza la fase de **diseño detallado**, que incluye

- **La definición de un modelo de agentes**, que identifica las clases de los agentes que implementarán el sistema y el número de instancias existentes de dichas clases.
- **La definición del modelo de servicios**, que identifica los principales servicios que han de cumplir los roles de los agentes y sus propiedades.

La metodología Gaia tiene un alcance limitado ya que excluye:

- Técnicas de modelado particulares. Se proponen algunas técnicas de modelado pero los autores, conscientes de la gran cantidad de trabajo existente en este aspecto, no se decantan por ninguna técnica en particular.
- Cuestiones de implementación. La salida de la metodología Gaia es una especificación detallada independiente de la tecnología usada para implementar el sistema multiagente, que debe ser fácilmente implementada usando una plataforma adecuada para la programación de agentes.
- Fase de requisitos. Los autores comprenden la importancia de esta fase y piensan que Gaia puede ser fácilmente integrada con cualquier sistema moderno orientado a un objetivo.

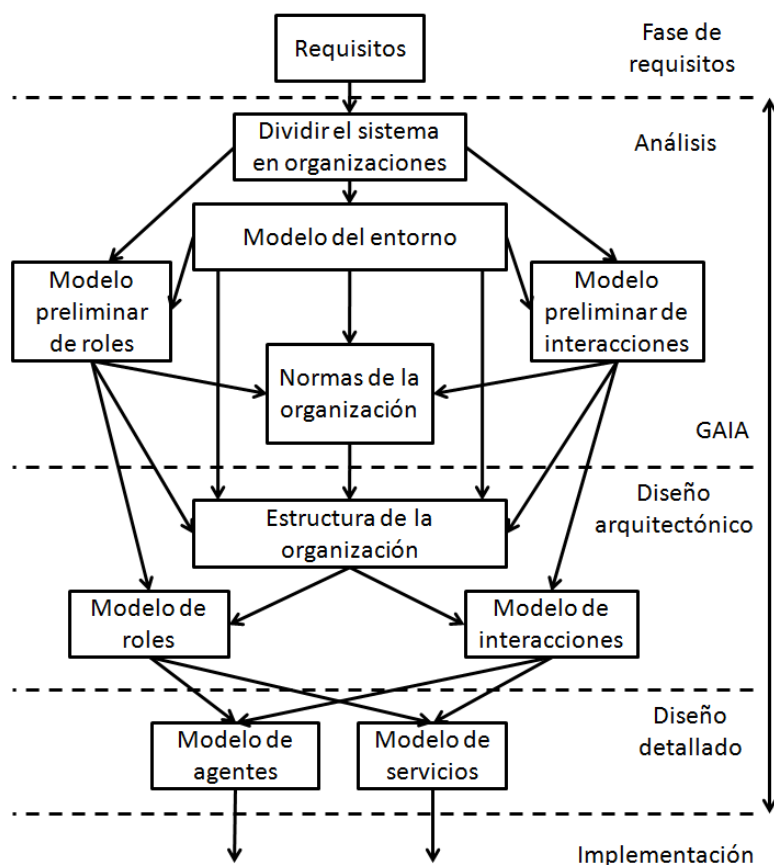


Ilustración 13 : Fases de la metodología Gaia II

## 2.5. Plataformas

Se han desarrollado muchas plataformas con la finalidad de permitir la interconexión y ejecución de los agentes de un sistema multiagente. Algunas siguen el estándar FIPA y otras se basan en arquitecturas de agente propias.

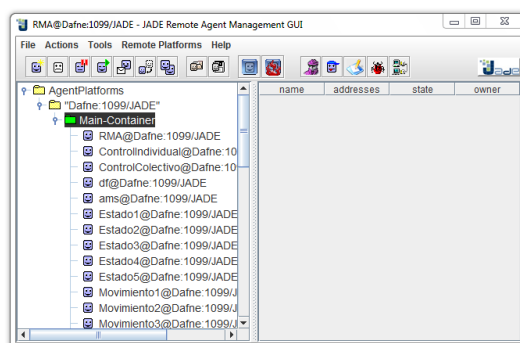
*Java Agent Template Lite (JatLite)*, está orientada a los aspectos de comunicación entre agentes. Las plataformas *Aglets* de IBM, y *Grasshopper* de IKV++, son soluciones para agentes móviles conformes al estándar MASIF de OMG. AgentBuilder, de Reticular Systems, es un entorno de desarrollo integrado que permite construir agentes BDI en Java. También ha tenido impacto el proyecto ZEUS, que ofrece una biblioteca de componentes Java junto a un sistema de planificación, varios protocolos de interacción y un conjunto de herramientas para construir agentes. Entre las plataformas que siguen el estándar de FIPA la más conocidas son *Java Agent DEvelopment Framework (JADE)*, *JADEX*[31] y *Jason*[32].

### 2.5.1. JADE

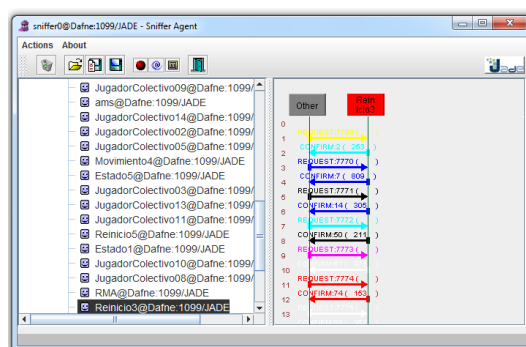
La plataforma JADE [13] es la implementación más extendida del estándar de FIPA[17]. Proporciona en primer lugar un conjunto de librerías para crear agentes que puedan comunicarse entre sí utilizando el lenguaje FIPA ACL (*Agent Communication Language*) y acceder a los servicios estándar de gestión de agentes FIPA.

Además JADE proporciona varios agentes ya implementados, que pueden ser utilizados para distintos propósitos, y que están ligados a la plataforma.

- El **Directory Facilitator** (DF) proporciona los servicios de directorio y puede conectarse a otros agentes del mismo tipo para gestionar, si fuera preciso, una estructura jerárquica de dominios.
- El **Agent Management System** (AMS) ejerce de supervisor de acceso y uso de la plataforma. El AMS proporciona un servicio de páginas blancas y de ciclo de vida, manteniendo una lista de los identificadores de los agentes y del estado en el que están. En la ilustración 15 podemos observar la interfaz de usuario que proporciona JADE para este servicio.
- El agente **Sniffer** ayuda en la depuración de las interacciones entre agentes. Este agente puede interceptar mensajes entre los distintos agentes y mostrarlos gráficamente para poder depurar sociedades de agentes, viendo el intercambio de mensajes entre ellos. En la ilustración 16 podemos observar la interfaz de usuario que proporciona JADE para este agente.
- El agente **Introspector** permite controlar el ciclo de vida de un agente en ejecución y los mensajes que intercambia, tanto los que haya en la cola de entrada o como en la de salida.



**Ilustración 14 : Remote Agent Management GUI de JADE**



**Ilustración 15 : Agente Sniffer de JADE**

Para implementar un agente en JADE contiene clases que proporcionan métodos para iniciar el agente en la plataforma con un nombre determinado, así como para mandar y recibir mensajes de otros agentes ya sea desde la misma y otras plataformas conformes al estándar de FIPA.

El contenido de los mensajes se describe con el lenguaje SL-0. Además, se pueden definir ontologías para enviar mensajes de acuerdo con un lenguaje y ontología particulares. De manera añadida, al crear una ontología se crean automáticamente un codificador y decodificador del contenido del mensaje según esa ontología particular. De esta forma, el implementador no tiene que preocuparse de cómo convertir a texto una determinada información o cómo analizar el mensaje. JADE se encarga de crear las clases JAVA con la información, y al definir la ontología se indican las clases que contienen la información que se desea enviar y recibir para que la plataforma cree los codificadores y decodificadores apropiados.

JADE también proporciona implementaciones de los protocolos estándar identificados en FIPA, protocolos que se definen en función de mensajes asíncronos como los descritos anteriormente. Se espera de los agentes que utilicen estos protocolos para preguntar a otros y

obtener la información, para negociar, para pedirles acciones, etc. El uso de los protocolos permite reutilizar gran cantidad de código en el desarrollo de los agentes.

Para integrar todo lo anterior con los distintos comportamientos que puede requerir un agente, JADE proporciona una arquitectura de agente básica. El acceso a los recursos y servicios de la plataforma se realiza a través de código encapsulado en clases predefinidas de JADE, que proporcionan un método para describir la tarea asignada a cada comportamiento. Estos comportamientos permiten enviar o recibir mensajes y organizar las actividades del agente en bloques de código. La idea es que todo el comportamiento del agente se codifique en estas clases.

### 2.5.2. JADEX

JADE proporciona una plataforma de agentes que facilita tanto las comunicaciones como la planificación de comportamientos, lo que es suficiente para usar agentes reactivos. Sin embargo, JADE no ofrece ninguna estructura de razonamiento interno, lo cual es necesario para usar agentes deliberativos.

JADEX es una extensión de JADE que permite usar agentes deliberativos en JADE. Los agentes JADEX se definen con dos elementos:

- Una descripción en XML dónde se incluye la información sobre las creencias, los deseos, planes, mensajes y estados iniciales del agente
- Una implementación en clases java de cada plan del agente, que puede acceder y modificar las creencias, satisface o abandona objetivos y adopta nuevos objetivos.

### 2.5.3. Jason

Jason (*Java-based AgentSpeak interpreter used with SACI for multi-agent distribution Over the Net*) es un interprete para una versión extendida de AgentSpeak. Jason adapta AgentSpeak para el desarrollo de sistemas multiagente, añadiendo para ello varias extensiones destinadas a crear un lenguaje de programación más práctico.

## 2.6. Aprendizaje por refuerzo

El **aprendizaje por refuerzo** [19][21] es una técnica de aprendizaje automático que se aplica principalmente en entornos muy activos y dinámicos. En aprendizaje por refuerzo, el agente percibe el entorno que tiene a su alrededor y ejecuta acciones que lo modifican en un proceso iterativo de ensayo y error. Cada vez que el agente ejecuta una acción sobre el entorno, éste le informa de su nuevo estado y de un refuerzo, que es alguna medida de la distancia al objetivo.

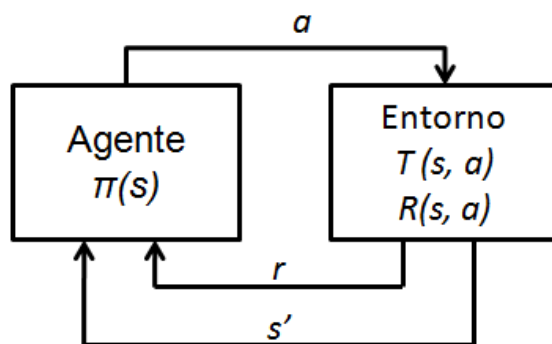


Ilustración 16 : Modelo de aprendizaje por refuerzo

El objetivo de las técnicas de aprendizaje por refuerzo es adquirir un conocimiento acerca del entorno que permita posteriormente elegir qué acción realizar en cada estado. El conocimiento del agente que podemos observar en la ilustración 16 normalmente está representado por una **política**  $\pi(s)$  que asocia a cada **estado**  $s$  con una **acción**  $a$ , por lo que el objetivo es encontrar una política que, de alguna forma, **maximice el refuerzo**  $r$  recibido a largo plazo.

El entorno que podemos observar en la ilustración 16 está definido por una **función de transición**  $T(s, a)$ , que asocia la **acción**  $a$  ejecutada en un estado  $s$ , con el estado al que transitará el agente  $s'$  y una **función de refuerzo**  $R(s, a)$  que asocia la acción  $a$  ejecutada en un estado  $s$  con el refuerzo  $r$  que recibirá el agente que ejecute dicha acción.

El entorno será no determinista si la función de transición no es determinista, esto es, si asocia el valor de la función de transición  $T(s, a)$  a una probabilidad. Será determinista si la misma acción  $a$  ejecutada en el mismo estado  $s$  da como resultado siempre el mismo estado  $s'$ .

### 2.6.1. Procesos de decisión de Markov

Las técnicas de aprendizaje por refuerzo basan la obtención de su conocimiento en ejecutar acciones que dependen únicamente del estado actual que el agente percibe en ese momento, lo que formalmente se conoce como **propiedad de Markov**. Así, la forma típica para definir las técnicas de aprendizaje por refuerzo es mediante **procesos de decisión de Markov**.

En los procesos de decisión de Markov se asume que:

- Existe un conjunto de **estados**.



- Existe una **función de transición** entre ellos, de forma que a un estado de partida y a una acción realizada en ese estado les asocia el estado en el que se encontrará el entorno después de ejecutar dicha acción.
- Existe una **función de refuerzo** que el entorno puede devolver al realizar la acción determinada en un entorno determinado.

En estos problemas se cumple la propiedad de Markov, por la que la política se basa solamente en el conocimiento del estado actual y no en ninguno anterior.

Formalmente, un proceso de decisión de Markov se define según la tupla  $\langle S, A, T, R \rangle$  tal que

- $S$  es un conjunto de estados.
- $A$  es un conjunto de acciones.
- $T: S \cdot A \rightarrow P(S)$  transforma estados en probabilidades, de forma que  $T(s, a, s')$  es la probabilidad de que estando en el estado  $s$  y ejecutando la acción  $a$  se transite al estado  $s'$ .
- $R: S \cdot A \rightarrow R$  proporciona el refuerzo  $r$  asociado a cada par estado-acción.

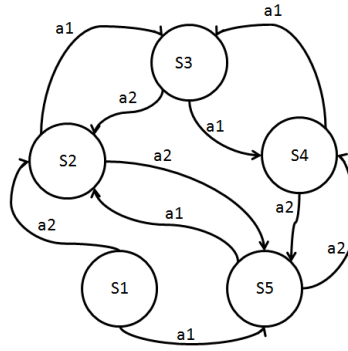
El agente debe encontrar una **política**  $\pi(s)$  que decida la acción a ejecutar en cada estado, considerando que la suma de los refuerzos recibidos a largo plazo debe ser maximizada. Este requisito de optimización se denomina **criterio de optimalidad**.

Un ejemplo de criterio de optimalidad es el **criterio de horizonte infinito descontado**, que tiene como objetivo maximizar la suma de los refuerzos  $r_k$  recibidos a lo largo del tiempo de la siguiente forma

$$\sum_{k=0}^{\infty} \gamma^k r_k$$

donde se introduce el parámetro de descuento  $\gamma$  que toma valores en  $[0,1]$ , reduciendo la influencia de los refuerzos obtenidos en el futuro.

Un problema de aprendizaje por refuerzo se denomina proceso de decisión de Markov si satisface la propiedad Markov, es decir, la toma de decisión sólo depende del estado actual. Por ejemplo, en el siguiente proceso de decisión de Markov finito y determinista



**Ilustración 17 : Ejemplo de proceso de decisión de Markov**

tiene las siguientes características asociadas:

- El conjunto de estados es  $S \equiv \{S1, S2, S3, S4, S5\}$
- El conjunto de acciones es  $A \equiv \{A1, A2\}$
- La función de transición asociada es

$T(s, a)$	S1	S2	S3	S4	S5
A1	S5	S3	S4	S3	S2
A2	S2	S5	S2	S5	S4

**Tabla 1 : Función de transición del proceso de decisión de Markov**

- Y la función de refuerzo  $R(s, a)$  tiene un valor de 1 para los pares (S3,A1) y (S5,A2) puesto que S4 es el estado final y éstas son las dos únicas transiciones que conducen a él.

### 2.6.2. Funciones de valor

Las funciones de valor son utilizadas por un gran número de algoritmos de aprendizaje por refuerzo para **aprender la política  $\pi$** . Estas funciones estiman cómo de bueno es para un agente estar en un estado determinado o cómo de bueno es usar una acción desde un determinado estado utilizando una política  $\pi$ . Se denota con  $V^\pi(s)$  al refuerzo que se espera obtener si comenzamos a guiarnos por la política  $\pi$  desde el estado  $s$  hasta el infinito. La definición formal para un MDP es la siguiente

$$V^\pi(s) = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s \right\}$$

$E_\pi$  denota el valor esperado cuando el agente sigue la política  $\pi$ . A  $V^\pi$  a la que se le denomina función de valor-estado. En la ecuación se introduce el factor de descuento del horizonte infinito  $\gamma$  definido anteriormente.

También podemos definir  $Q^\pi(s, a)$  como el valor al ejecutar una acción  $a$  desde un estado  $s$  siguiendo una política  $\pi$ . En este caso a  $Q^\pi$  se le denomina función de valor-acción. Su definición formal es la siguiente:

$$Q^\pi(s, a) = E_\pi \{R_t | s_t = s, a_t = a\} = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s | a_t = a \right\}$$

Dentro del conjunto de posibles políticas existe una serie de políticas que son mejores o iguales que el resto, a las que se las denomina políticas óptimas  $\pi^*$ . Estas políticas óptimas comparten una única función de valor-estado que maximiza el valor para cualquier política, denominada función valor-estado óptima  $V^*$

$$V^*(s) = \max_{\pi} V^\pi(s), \forall s \in S$$

De igual forma se define la función valor-acción óptima  $Q^*(s, a)$ , que maximizará el valor de cualquier par estado-acción para cualquier política

$$Q^*(s, a) = \max_{\pi} Q^\pi(s, a), \forall s \in S, \forall a \in A$$

Partiendo de las funciones de valor óptimas es fácil obtener políticas óptimas, usando las acciones que maximizan el valor de  $Q^*(s, a)$  para cada estado  $s$  de la forma

$$\pi^*(s) = \arg_a \max Q^*(s, a)$$

### 2.6.3. Métodos de resolución

Existen diversos métodos de resolución para los problemas de aprendizaje por refuerzo que se basan en el mantenimiento de las funciones de valor para estructurar el conocimiento. Obteniendo los valores óptimos de estas funciones se obtiene una representación óptima de la política a seguir. Las funciones óptimas cumplen las ecuaciones de optimalidad de Bellman que formulan matemáticamente el principio de optimalidad por el que dado un estado, la política óptima para las siguientes etapas no depende de la política tomada en las etapas anteriores. De esta forma, la decisión óptima inmediata sólo depende del estado en el que se está y no de cómo se llegó hasta él. Toda la información sobre el pasado se resume en el estado en el que se encuentra.

Sólo es posible usar técnicas de programación dinámica cuando se posee un conocimiento completo del entorno como un Proceso de Decisión de Markov. Es decir, cuando se conocen los conjuntos de estados y acciones y las funciones de transición de estados y de refuerzo.

En la práctica no se suele disponer de dicho conocimiento ya que en la mayoría de los casos sólo disponemos de un conocimiento parcial del problema. Ante esta situación existe la posibilidad de aprender primero la dinámica del problema mediante exploración para después emplear las técnicas de programación dinámica. Estos métodos son conocidos como *Métodos basados en el modelo*, que tienen que aprender.

Cuando el desconocimiento del problema es total se usan los *Métodos libres del modelo*, que basan el cálculo de las funciones de valor en la experiencia obtenida durante la interacción con el entorno, sin necesidad de tener un conocimiento explícito de la dinámica del mismo. Dichos métodos son los que mejor se adaptan a los problemas reales.

#### **2.6.4. Métodos libres de modelo**

En los métodos libres de modelo se asume un total desconocimiento de la dinámica del entorno, es decir, se conocen los conjuntos de estados y de acciones, pero no cuál es el efecto de cada acción en cada estado.

Por tanto, los métodos libres de modelo se basan sólo en el conocimiento que obtienen de ejecutar acciones sobre el entorno y observar sus consecuencias. El conocimiento se puede adquirir de dos formas distintas:

- De secuencias de acciones completas ejecutadas por el agentes desde el estado inicial hasta el estado final. Por ejemplo, los métodos de Monte Carlo.
- De acciones concretas ejecutadas por el agente en un estado concreto. Por ejemplo, los métodos de diferencia temporal.

Todos estos métodos se basan en las funciones de valor definidas anteriormente para representar las políticas de acción.

### **2.7. Q-Learning**

El algoritmo de Q-Learning [19] es un *método libre del modelo* que aprende a partir de conocimiento adquirido del entorno en cada acción ejecutada en cada estado, es decir, a partir de la experiencia generada durante la exploración del entorno.

Esta experiencia se representa mediante tuplas de experiencia  $\langle s, a, s', r \rangle$  que contienen

- $s$ , el estado actual en el que se encontraba el agente al ejecutar la acción.
- $a$ , a acción ejecutada por el agente.
- $s'$ , el estado al que se ha llegado ejecutando la acción.
- $r$ , el refuerzo obtenido por ello.

Dicha experiencia se encuentra en la función  $Q(s, a)$  que mapea un estado y una acción ejecutada con un valor de la calidad de dicha acción en dicho estado. De esta forma el algoritmo de Q-learning es el siguiente:

```

• Inicializar  $Q(s, a)$  arbitrariamente.
• Repetir (episodio de aprendizaje)
    o Inicializar el estado  $s$ .
    o Repetir (ciclo de aprendizaje)
        ▪ Seleccionar una acción  $a$  a partir de  $s$ , usando alguna
          política derivada de  $Q(s, a)$ .
        ▪ Ejecutar dicha acción  $a$ , recibiendo el refuerzo  $r$  y el
          siguiente estado  $s'$ .
        ▪ Actualizar la entrada de la tabla  $Q(s, a)$  según la ecuación
           $Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha [r_{s,a} + \gamma \max_{a'} Q(s', a')]$ 
        ▪  $s \leftarrow s'$ 
    o Hasta que  $s$  sea un estado final.
```

**Ilustración 18 : Algoritmo de Q-learning**

En este algoritmo se ha incluido el parámetro de enfriamiento  $\alpha$ , que valora la importancia de las nuevas actualizaciones frente a los valores ya existentes. El parámetro de enfriamiento  $\alpha$  es sólo necesario cuando el dominio es estocástico. Si el dominio fuera determinista, es decir, si el efecto de una acción en un estado es siempre el mismo,  $\alpha$  puede tomar un valor de 1, quedando la ecuación en

$$Q(s, a) \leftarrow r_{s,a} + \gamma \max_{a'} Q(s', a')$$

Para nuestro ejemplo, asumiremos que la tabla  $Q(s, a)$  inicial del proceso de decisión de Markov es

$Q(s, a)$	$a_1$	$a_2$
$s_1$	0	0
$s_2$	0	0
$s_3$	0	0
$s_4$	0	0
$s_5$	0	0

Supongamos que en el primer episodio de aprendizaje se ejecuta la secuencia de acciones

$$S_1 \xrightarrow{a_2} S_2 \xrightarrow{a_1} S_3 \xrightarrow{a_1} S_4$$

Ello dará lugar a las siguientes actualizaciones de la tabla Q.

$$Q(S_1, a_1) = r_{S_1, a_1} + \gamma \max_a Q(S_2, a) = 0 + \gamma 0 = 0$$

- $Q(S_1, a_2) = r_{S_1, a_2} + \gamma \max_a Q(S_2, a) = 0 + \gamma 0 = 0$
- $Q(S_2, a_1) = r_{S_2, a_1} + \gamma \max_a Q(S_3, a) = 0 + \gamma 0 = 0$
- $Q(S_3, a_1) = r_{S_3, a_1} + \gamma \max_a Q(S_4, a) = 1 + \gamma 0 = 1$

Por lo que la tabla Q quedará en la forma

$Q(s, a)$	$a_1$	$a_2$
$S_1$	0	0
$S_2$	0	0
$S_3$	1	0
$S_4$	0	0
$S_5$	0	0

Supongamos que en el segundo episodio se ejecuta la secuencia de acciones

$$S_1 \xrightarrow{a_1} S_5 \xrightarrow{a_1} S_2 \xrightarrow{a_1} S_3 \xrightarrow{a_1} S_4$$

Ello dará lugar a las siguientes actualizaciones de la tabla Q.

- $Q(S_1, a_1) = r_{S_1, a_1} + \gamma \max_a Q(S_5, a) = 0 + \gamma 0 = 0$
- $Q(S_5, a_1) = r_{S_5, a_1} + \gamma \max_a Q(S_2, a) = 0 + \gamma 0 = 0$
- $Q(S_2, a_1) = r_{S_2, a_1} + \gamma \max_a Q(S_3, a) = 0 + \gamma 1 = \gamma$
- $Q(S_3, a_1) = r_{S_3, a_1} + \gamma \max_a Q(S_4, a) = 1 + \gamma 0 = 1$

Por lo que la tabla Q quedará en la forma

$Q(s, a)$	$a_1$	$a_2$
$S_1$	0	0

$S_2$	$\gamma$	0
$S_3$	1	0
$S_4$	0	0
$S_5$	0	0

Supongamos que en el tercer episodio se ejecuta la secuencia de acciones

$$S_1 \xrightarrow{a_2} S_5 \xrightarrow{a_2} S_4$$

Ello dará lugar a las siguientes actualizaciones de la tabla Q.

- $Q(S_1, a_1) = r_{S_1, a_1} + \gamma \max_a Q(S_5, a) = 0 + \gamma 0 = 0$
- $Q(S_5, a_2) = r_{S_5, a_2} + \gamma \max_a Q(S_4, a) = 1 + \gamma 0 = 1$

Por lo que la tabla Q quedará en la forma

$Q(s, a)$	$a_1$	$a_2$
$S_1$	0	0
$S_2$	$\gamma$	0
$S_3$	1	0
$S_4$	0	0
$S_5$	0	1

Ejecutando un número de episodios suficientemente extenso el algoritmo converge a la tabla Q óptima siguiente

$Q(s, a)$	$a_1$	$a_2$
$S_1$	$\gamma$	$\gamma^2$
$S_2$	$\gamma$	$\gamma$
$S_3$	1	$\gamma^2$
$S_4$	0	0
$S_5$	$\gamma^2$	1

Tabla 2 : Tabla Q óptima del ejemplo

De esta tabla Q óptima se puede obtener la política óptima para llegar de  $S_1$  a  $S_4$  es:

$$S_1 \xrightarrow{a_2} S_5 \xrightarrow{a_2} S_4$$

### 2.7.1. Exploración y explotación

En el aprendizaje por refuerzo se entiende por *explorar* la ejecución de acciones según una política de decisión que permita visitar nuevos estados y estudiar nuevas alternativas para alcanzar el objetivo del problema. En oposición, por *explotación* se entiende el hecho de utilizar el conocimiento adquirido durante el proceso de aprendizaje para obtener un máximo beneficio.

El equilibrio entre explotación y exploración puede alcanzarse mediante diferentes estrategias que, en mayor o menor medida, producen acciones aleatorias:

- Una primera estrategia sería la de seleccionar siempre una acción aleatoria. Ello implica que se puede elegir con igual probabilidad la mejor acción y la peor.
- Una segunda estrategia sería la de seleccionar siempre la mejor acción conocida para un estado. Este tipo de estrategia es conocida como *greedy* e impide explorar nuevas soluciones cuando se conoce una que es, posiblemente, peor.

Existen estrategias mixtas como la estrategia  $\epsilon$  – *greedy* que selecciona una acción aleatoria con probabilidad  $\epsilon$  y una acción conocida con probabilidad  $1 - \epsilon$ .

La estrategia  $\epsilon$  – *greedy* obtiene muy buenos resultados al principio del proceso de aprendizaje, cuando las soluciones conocidas están muy lejos de ser óptimas y hace falta seguir explorando el entorno para encontrar soluciones mejores. Sin embargo, es posible que en entornos estocásticos, cuando la solución óptima es conocida, la elección de una acción



aleatoria en un momento determinado puede hacer que el agente no consiga alcanzar su objetivo en un episodio. Por ello, a veces son necesarias estrategias que varíen la probabilidad de decisión entre una acción aleatoria y una acción conocida.

### 2.7.2. Aproximación de funciones

El algoritmo de Q-Learning se basa en estimar la función de valor-acción,  $Q(s,a)$ , mediante la experiencia acumulada en los distintos episodios de aprendizaje. Habitualmente la representación de la función  $Q(s,a)$  es una tabla de valores. Sin embargo, es posible aproximarla mediante técnicas de aproximación de funciones como redes de neuronas o árboles de decisión.

El hecho de que la función  $Q(s,a)$  dependa de dos parámetros, estado  $s$  y acción  $a$ , da lugar a dos alternativas posibles a la hora de aproximar su valor:

- En la primera alternativa, se usa un aproximador  $\hat{Q}(s,a)$  que toma como entrada tanto el estado  $s$  como la acción  $a$  obteniendo un valor  $q$ .
- En la segunda alternativa, si el conjunto de acciones es reducido, se puede usar un aproximador de  $Q(s,a)$  para cada posible acción  $a$ , obteniendo tantos aproximadores como acciones sean posibles,  $\hat{Q}_{ai}(s)$ .

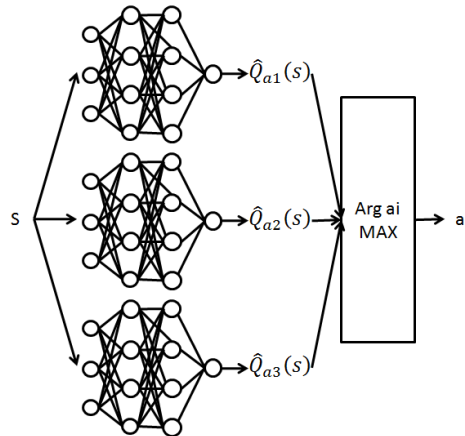


Ilustración 19 : Aproximación con redes de neuronas de la función  $Q(s,a)$  [21]

## 2.8. Aprendizaje por refuerzo en sistemas multiagente

Además de los beneficios propios de la naturaleza distribuida de los sistemas multiagente, como el incremento de la eficiencia posible gracias al paralelismo, el uso de múltiples agentes implementando técnicas de aprendizaje por refuerzo puede explotar nuevos beneficios, por ejemplo, compartiendo experiencia, comunicándose, aprendiendo o imitándose entre ellos.

Por el contrario, además de los desafíos planteados por el aprendizaje por refuerzo monoagente, como el equilibrio entre exploración y explotación, nuevos desafíos surgen en el aprendizaje por refuerzo multiagente, como la dificultad de especificar un aprendizaje objetivo, los problemas de aprendizaje pasan a ser no estacionarios o la necesidad de coordinar a los distintos agentes.

### **2.8.1. Beneficios del aprendizaje por refuerzo multiagente**

El incremento de la eficiencia puede obtenerse gracias al paralelismo cuando los agentes explotan estructuras descentralizadas de la tarea a ejecutar. Dividiendo correctamente la tarea en subtareas y asignándolas a distintos agentes paralelos se incrementará la eficiencia global del sistema.

El hecho de compartir experiencia puede ayudar a que agentes con tareas similares puedan aprender más rápido y mejor. Por ejemplo:

- Los agentes pueden intercambiar información mediante comunicación directa. De forma que un agente puede aprender de su propia experiencia y además de la de otros agentes.
- Los más habilidosos en sus tareas pueden enseñar al resto cómo solucionar sus tareas.
- El agente que todavía está aprendiendo puede observar e imitar a los agentes más habilidosos, valiéndose nuevamente de experiencias de otros agentes para acelerar su proceso de aprendizaje.

Además, cuando en un sistema multiagente uno o más agentes fallan en su tarea, el resto de agentes puede ocuparse de sus tareas, lo que hace el sistema multiagente más robusto. La mayoría de las plataformas de agentes permiten añadir fácilmente nuevos agentes en el sistema.

### **2.8.2. Desafíos en aprendizaje por refuerzo multiagente**

El principal reto en aprendizaje por refuerzo multiagente estriba en el crecimiento exponencial del espacio en número de estados, dado que los algoritmos básicos de aprendizaje por refuerzo, como Q-Learning, estiman valores para cada estado discreto posible, o par estado-acción. Se produce un incremento exponencial de la complejidad computacional en sistemas multiagente puesto que cada agente ha de tener en cuenta, además de su propio estado, el estado de otros agentes o, al menos, mantener alguna información acerca de ellos. La complejidad también crece exponencialmente con el número de agentes, porque cada agente añade sus propias variables al espacio de pares estado-acción compartido.

Especificar una buena meta en aprendizaje por refuerzo multiagente es también un reto, puesto que los refuerzos de los agentes están correlacionados y no pueden maximizar independientemente el refuerzo adquirido. El hecho de que un agente intente maximizar el refuerzo individual recibido puede hacer que el refuerzo global de todos los agentes se decremente, por lo que es indispensable un buen diseño de la meta.

La naturaleza no estacionaria del sistema multiagente surge porque todos los agentes del sistema están aprendiendo al mismo tiempo. Cada agente se enfrenta a un problema de aprendizaje cuya solución está en movimiento puesto que los cambios en las políticas de otros agentes, con los que comparte el entorno, cambian la política del propio agente.

El problema del equilibrio entre exploración y explotación exige algoritmos que tengan en cuenta el hecho de que la explotación de un agente puede influir en la exploración de otro agente y viceversa. Demasiada exploración puede desestabilizar la dinámica de aprendizaje de los otros agentes, lo que hace el proceso de aprendizaje más difícil para el agente explorador. De igual forma, demasiada explotación del conocimiento adquirido puede hacer que otros agentes, con los que se coopera, se vean influenciados por una solución subóptima explotada.

La necesidad de coordinación surge del hecho de que cada acción tomada en el entorno depende de las acciones tomadas por otros agentes. Por lo tanto, las decisiones de los agentes deben ser coherentes entre sí para lograr el efecto pretendido. La coordinación es básica en configuraciones cooperativas, pero es también conveniente para agentes no cooperativos puesto que puede simplificar su aprendizaje al hacer más previsibles los efectos de sus acciones.

### **2.8.3. Clasificación para los algoritmos de aprendizaje por refuerzo multiagente**

Los algoritmos de aprendizaje por refuerzo multiagente pueden ser clasificados según varios criterios:

- **Homogeneidad.** Algunos algoritmos de aprendizaje por refuerzo multiagente sólo funcionan si todos los agentes implicados en el proceso utilizan el mismo algoritmo (team-Q, Nash-Q), mientras que otros permiten que haya agentes en el sistema aprendiendo mediante algoritmos diferentes (AWESOME[27], WoLF-PHC).
- **Conocimiento anterior de la tarea.** El modelo está disponible para el agente desde el principio (AWESOME) o bien es desconocido (Nash-Q[26], team-Q, WoLF-PHC).
- **Tipo de entrada que reciben los agentes.** Normalmente la entrada representa exactamente el estado del entorno, pero aparecen diferencias cuando en ella se incluyen las observaciones de otros agentes. Por ejemplo, un algoritmo puede necesitar observar acciones y los refuerzos de otros agentes (Nash-Q), sólo las

acciones de otros agentes (team-Q, AWESOME) o no necesitar ninguna información acerca de otros agentes (WoLF-PHC).

Según la cooperación necesaria para resolver el problema planteado a un sistema multiagente, existe una clasificación adicional que incluye tareas cooperativas y tareas competitivas.

#### 2.8.4. Tareas cooperativas

El algoritmo de Q-Learning distribuido [26] resuelve el problema de la tarea cooperativa sin asumir ninguna coordinación entre los agentes y con una complejidad similar al algoritmo Q-Learning monoagente. En cambio, el algoritmo sólo funciona en problemas deterministas. Cada agente  $i$  mantiene una política local  $\bar{h}_i(x)$  y una función  $Q_i(x, u_i)$  local, que dependen únicamente de sus acciones. Los valores de la función  $Q_i$  local son actualizados solo cuando la actualización incrementa el valor de dicha función  $Q_i$ .

$$Q_{i,k+1}(x_k, u_{i,k}) = \max\{Q_{i,k}(x_k, u_{i,k}), r_{k+1} + \gamma \max_{u_i} Q_{i,k}(x_{k+1}, u_i)\}$$

Gracias a ello, el valor local de la función  $Q_i$  siempre contiene el máximo valor de la función conjunta  $Q_{i,k}(x_k, u_i) = \max_u Q_k(x, u)$ . La política local es actualizada sólo si la actualización supone una mejora en los valores de la función  $Q_i$ .

$$\bar{h}_{i,k+1}(x_k) \begin{cases} u_{i,k} & \text{si } \max_{u_i} Q_{i,k}(x_{k+1}, u_i) > \max_{u_i} Q_{i,k}(x_k, u_i) \\ \bar{h}_{i,k}(x_k) & \text{en cualquier otro caso} \end{cases}$$

Así se asegura que la política conjunta  $[\bar{h}_{1,k}, \dots, \bar{h}_{n,k}]^T$  sea siempre óptima respecto a la función  $Q_k$  global. Bajo la condición de que  $Q_{i,0} = 0 \forall i$ , las políticas locales de los agentes probablemente converjan a una política óptima conjunta.

Otro ejemplo es el de *Coordination Graphs*[29], en el que se usa coordinación directa entre los agentes. Con *Coordination Graphs* se puede simplificar la coordinación, descomponiendo la función  $Q$  global en funciones  $Q_i$  locales que sólo dependen de las acciones de un subconjunto de los agentes. Por ejemplo, para un entorno con 4 agentes, la descomposición podría ser

$$Q(x, u) = Q_1(x, u_1, u_2) + Q_2(x, u_1, u_3) + Q_3(x, u_3, u_4)$$

De esta forma, las funciones  $Q_i$  locales serán mucho más manejables que una función  $Q$  global. Además, maximizando las funciones  $q$  locales, se conseguirá maximizar la función  $Q$  global.

### 2.8.5. Tareas competitivas

En las tareas competitivas el refuerzo positivo de un agente es el refuerzo negativo del otro y viceversa. En ellas, el principio del algoritmo *minimax* puede ser aplicado, es decir, maximizar el beneficio propio suponiendo que el oponente va a intentar siempre minimizarlo.

El algoritmo *minimax-Q* [30] emplea el principio *minimax* para calcular estrategias y un algoritmo similar a *Q-Learning* para propagar los refuerzos obtenidos entre los pares estado-acción. Para el primer agente

$$h_{1,k}(x_k) = \arg m_1(Q_k, x_k)$$

$$Q_{k+1}(x_k, u_{1,k}, u_{2,k}) = (1 - \alpha)Q_k(x_k, u_{1,k}, u_{2,k}) + \alpha [r_{k+1} + \gamma m_1(Q_k, x_{k+1})]$$

Siendo  $m_1$  el valor que *minimax* devuelve para el agente 1

$$m_1(Q, x) = \max_{h_1(x_1)} \min_{u_2} \sum_{u_1} h_1(x, u_1) Q_k(x, u_1, u_2)$$

## 2.9. Conclusiones y visión general del sistema

El presente proyecto implementa un sistema multiagente según la metodología Gaia en la plataforma para la ejecución de agentes JADE.

Se ha decidido usar Gaia[6] en su segunda versión debido a que es una metodología en la que resulta realmente atractivo el tratamiento de los agentes tanto desde un aspecto general (sociedades de agentes) como desde un aspecto concreto definiendo agentes y servicios. Se trata además de una metodología muy abierta que no se centra en ningún tipo de arquitectura.

Por otra parte, la elección de JADE se debe a que es la plataforma más extendida de todas aquellas que implementan el estándar FIPA[17]. Además, por el hecho de estar implementada en Java, facilita el desarrollo de programas y la inclusión en un futuro de herramientas ya implementadas en este lenguaje.

A continuación se procede a definir el problema que ha de resolver el sistema, así como los recursos existentes y el tipo de aprendizaje a realizar.

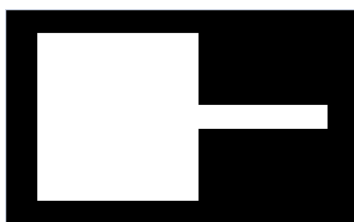
### 2.9.1. Entorno del problema a resolver

El entorno en el que se encuadra el problema a resolver es una cuadrícula, llamada tablero, en la que cada una de sus casillas sólo puede encontrarse

- Ocupada por un obstáculo,
- Ocupada por un único agente o
- Libre.

Se trata por tanto de un entorno discreto en el que los agentes ocupan un, y sólo un, estado. Las transiciones deben realizarse atómicamente de estado a estado, no siendo posible que un agente se encuentre en transición entre estados.

En el siguiente ejemplo, todas las casillas están libres (en blanco) o son obstáculos (en negro).

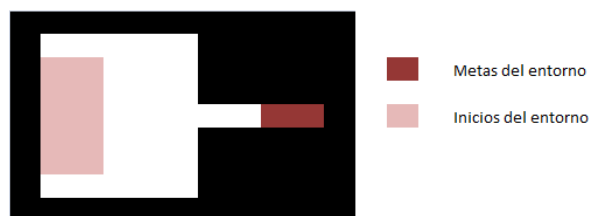


**Ilustración 20 : Ejemplo de obstáculos en un entorno**

Los obstáculos del entorno serán siempre fijos, no pudiendo modificarse ni moverse en ningún momento. Además, algunas casillas podrán tener una o las dos siguientes propiedades.

- Ser una de las casillas en las que los agentes empiezan su ejecución
- Ser una de las casillas objetivo del problema

Al menos debe existir una casilla de cada tipo, pudiendo existir tantas como casillas libres haya en el entorno. En el siguiente ejemplo, las casillas rosas son los inicios de los agentes y las casillas granate son las casillas meta.



**Ilustración 21 : Casillas de inicio y casillas meta en un entorno**

El entorno del problema a resolver puede requerir que los agentes se dividan en varios equipos, teniendo cada equipo sus propios inicios y metas. El siguiente tablero está preparado para dos equipos:



**Ilustración 22 : Ejemplo de entorno de dos equipos**

Los agentes podrán desplazarse dentro del entorno mediante 5 diferentes acciones:

- Desplazarse hacia el norte.
- Desplazarse hacia el este.
- Desplazarse hacia el sur.
- Desplazarse hacia el oeste.
- No desplazarse.

### **2.9.2. Problema a resolver**

El problema a resolver será el de diseñar e implementar un sistema multiagente capaz de encontrar un conjunto de políticas de acción  $\pi_i(s) \rightarrow a$  que asigne a cada posible estado  $s$  del tablero una acción  $a$  para cada equipo  $i$ .

Durante la resolución del problema, los agentes compartirán el tablero de juego con el resto de agentes que estén intentando resolver el problema, por lo que el entorno no es exclusivo.

### **2.9.3. Recursos existentes en el sistema**

El sistema incluirá la programación de los siguientes recursos:

- 1) Un tablero de juego en el que se representará el problema a solucionar y en el que el usuario podrá observar en todo momento la situación de los agentes que tratan de resolverlo.
- 2) Un gestor estadístico en el que los agentes depositarán las estadísticas de sus procesos de aprendizaje para ser mostradas al usuario.
- 3) Un repositorio del conocimiento colectivo que los agentes podrán usar para obtener los mejores movimientos según distintas políticas.

Todos los recursos del sistema serán gestionados por agentes del propio sistema, que servirán dichos recursos a los agentes del sistema encargados de obtener la solución del

problema. La cantidad de agentes gestores, así como la capacidad de cada uno de ellos, será configurable desde la interfaz de usuario.

#### **2.9.4. Proceso de aprendizaje**

En el proceso de aprendizaje se usará el algoritmo de aprendizaje por refuerzo Q-Learning, con las siguientes características:

- Cada agente que implemente el algoritmo de aprendizaje deberá afrontar el problema de forma distribuida, por lo que
  - No podrá ejecutar sus acciones directamente sobre el entorno, deberá valerse de los servicios de otro agente que ejecute sus acciones.
  - No podrá recibir la información acerca del estado en el que se encuentra o el refuerzo recibido directamente del entorno, deberá valerse de los servicios de otro agente para obtener su información.
  - Podrá ser interferido en sus acciones por otros agentes.
  - Podrá obtener la experiencia de otros agentes que intentan resolver el mismo problema para la obtención de su propia solución.
- El algoritmo deberá implementar una política  $\varepsilon - greedy$  para la decisión de la acción a tomar que equilibre en el tiempo la relación entre exploración y explotación.
- Usará la función de actualización del algoritmo de Q-Learning original para entornos no estacionarios:

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha [r_{s,a} + \gamma \max_{a'} Q(s', a')]$$

- Los agentes pueden valerse de un conocimiento previo del problema como semilla de su proceso de aprendizaje, siempre y cuando dicho conocimiento haya sido adquirido previamente por algún agente del propio sistema.
- Los agentes podrán compartir los conocimientos adquiridos.



### 3. Análisis del sistema

En este capítulo se pretende dar una visión general del sistema, analizando el algoritmo de aprendizaje, las organizaciones que estarán presentes en el sistema, el entorno de ejecución, los roles que podrán tomar los agentes y sus interacciones, además de las reglas que regirán los roles e interacciones. Este capítulo se corresponde al Análisis propuesto por la metodología Gaia, añadiendo los apartados necesarios para realizar dicho análisis sobre el proceso de aprendizaje.

#### 3.1. Descripción del proceso de aprendizaje

Para dotar al sistema multiagente de un cierto grado de inteligencia en la toma de sus decisiones es necesario implementar en él algún tipo de técnica de inteligencia artificial. Debido a la autonomía de los agentes, a la división del entorno tablero en casillas, a la posibilidad de interactuar con dicho entorno y a los tipos de movimientos usados, es un problema adecuado para usar algoritmos de **aprendizaje por refuerzo**.

##### 3.1.1. Algoritmo usado en el proceso de aprendizaje

El algoritmo usado para todos los procesos de aprendizaje es el algoritmo de Q-Learning, que se describe como

```
- Inicializar  $Q(s,a)$  arbitrariamente.
- Repetir (episodio de aprendizaje)
  o Inicializar el estado  $s$ .
  o Repetir (ciclo de aprendizaje)
    ▪ Seleccionar una acción  $a$  a partir de  $s$ , usando alguna política derivada de  $Q(s,a)$ .
    ▪ Ejecutar dicha acción  $a$ , recibiendo el refuerzo  $r$  y el siguiente estado  $s'$ .
    ▪ Actualizar la entrada de la tabla  $Q(s,a)$  según la ecuación
      
$$Q(s,a) \leftarrow (1 - \alpha)Q(s,a) + \alpha [r_{s,a} + \gamma \max_{a'} Q(s',a')]$$

    ▪  $s \leftarrow s'$ 
  o Hasta que  $s$  sea un estado final.
```

**Ilustración 23 : Algoritmo de Q-learning inicial**

Como los agentes deben obtener su información a través de comunicaciones establecidas con otros agentes, el algoritmo debe ser adaptado. Además, para que su implementación se pueda aplicar a un agente, deben establecerse unos límites en la duración de los episodios y ciclos de aprendizaje. De esta forma el algoritmo modificado para su implementación queda en la forma

- Inicializar  $Q(s,a) = 0, \forall s,a$ .
- Repetir (episodio de aprendizaje)
  - o Reiniciar la posición en el entorno.
  - o Pedir y recibir información acerca del estado actual
  - o Repetir (ciclo de aprendizaje)
    - Seleccionar una acción  $a$ .
    - Ejecutar dicha acción  $a$ , recibiendo el refuerzo  $r$ .
    - Pedir y recibir información acerca del estado actual
    - Actualizar la entrada de la tabla  $Q(s,a)$ , según la ecuación
 
$$Q(s,a) \leftarrow (1 - \alpha)Q(s,a) + \alpha [r_{s,a} + \gamma \max_{a'} Q(s',a')]$$
    - $s \leftarrow s'$
  - o Hasta que  $s$  sea un estado final o se cumpla el límite de ciclos para un episodio.
  - o Informar de las estadísticas del episodio completado.
- Hasta que se cumpla el límite de episodios previsto.

Ilustración 24 : Algoritmo de Q-learning adaptado para establecer las comunicaciones necesarias

### 3.1.2. Control y coordinación del algoritmo de aprendizaje


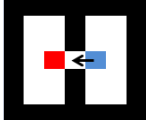
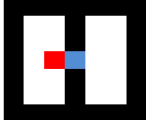
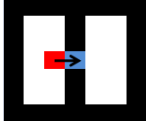
Además es necesario realizar una segunda modificación debida a que el agente se encuentra en un **entorno compartido** con otros agentes con quienes compite por su uso. Para el proceso de aprendizaje es básico que ningún agente ejecute dos ciclos si otro agente no ha llegado a ejecutar ninguno, ya que así se mantiene el equilibrio entre los procesos de aprendizaje de todos los agentes. De lo contrario se crearían condiciones de carrera por las cuales un agente podría ejecutar más ciclos que el resto si consiguiera acceder más veces al entorno.

Además, cada agente deberá coordinarse con otros agentes para ejecutar sus movimientos. La operación de movimiento se encuentra dentro del ciclo de aprendizaje, por lo que es necesario regular el acceso a dicho ciclo.

- Inicializar  $Q(s,a) = 0, \forall s,a$ .
- Repetir (episodio de aprendizaje)
  - o Reiniciar la posición en el entorno.
  - o Pedir y recibir información acerca del estado actual
  - o Repetir (ciclo de aprendizaje)
    - Informar de que el agente se encuentra listo para ejecutar un ciclo.
    - Esperar permiso para ejecutar el ciclo.
    - Seleccionar una acción  $a$ .
    - Ejecutar dicha acción  $a$ , recibiendo el refuerzo  $r$ .
    - Pedir y recibir información acerca del estado actual
    - Actualizar la entrada de la tabla  $Q(s,a)$ , según la ecuación
 
$$Q(s,a) \leftarrow (1 - \alpha)Q(s,a) + \alpha [r_{s,a} + \gamma \max_{a'} Q(s',a')]$$
    - $s \leftarrow s'$
    - Informar de que el agente ha acabado el ciclo.
  - o Hasta que  $s$  sea un estado final o se cumpla el límite de ciclos para un episodio
  - o Informar de las estadísticas del episodio completado.
- Hasta que se cumpla el límite de episodios previsto.

Ilustración 25 : Algoritmo de Q-learning adaptado para establecer las comunicaciones necesarias e interactuar con el agente de control en cada ciclo

Estas modificaciones son necesarias debido a que el entorno es discreto y las operaciones de transición entre estados son atómicas, además de que se supone un paralelismo. De no realizar estas modificaciones sería posible la siguiente situación

	<p>Dos agentes se encuentran en el entorno:</p> <ul style="list-style-type: none"> <li>El agente 1 (rojo) observa que la casilla situada al este se encuentra libre.</li> <li>El agente 2 (azul) observa que la casilla situada al oeste se encuentra libre</li> </ul>
	El agente 2 (azul) solicita moverse hacia el oeste, puesto que conoce que su situación es libre.
	Y realiza su movimiento sin ningún problema.
	El agente 1 (rojo) solicita moverse hacia el este, puesto que el conocimiento que tiene de ello es que la casilla está vacía, chocando contra el agente 2 (azul).

**Ilustración 26 : Ejemplo de interbloqueo entre agentes**

Para evitar estas situaciones, en las que el estado ha sido modificado en el intervalo entre la observación y la ejecución de una acción, es necesario este **control de acceso al ciclo de aprendizaje**.

Además es necesaria una tercera modificación para impedir que los agentes ejecuten episodios de aprendizaje desordenadamente. Para un correcto aprendizaje, es necesario que todos los agentes simultáneamente **ejecuten el mismo episodio**.

<ul style="list-style-type: none"> <li>- Inicializar <math>Q(s,a) = 0, \forall s,a</math>.</li> <li>- Informar de que el agente se encuentra listo para ejecutar un episodio.</li> <li>- Esperar permiso para ejecutar el episodio.</li> <li>- Repetir (episodio de aprendizaje) <ul style="list-style-type: none"> <li>o Reiniciar la posición en el entorno.</li> <li>o Pedir y recibir información acerca del estado actual</li> <li>o Repetir (ciclo de aprendizaje) <ul style="list-style-type: none"> <li>▪ Informar de que el agente se encuentra listo para ejecutar un ciclo.</li> <li>▪ Esperar permiso para ejecutar el ciclo.</li> <li>▪ Seleccionar una acción <math>a</math>.</li> <li>▪ Ejecutar dicha acción <math>a</math>, recibiendo el refuerzo <math>r</math>.</li> <li>▪ Pedir y recibir información acerca del estado actual</li> <li>▪ Actualizar la entrada de la tabla <math>Q(s,a)</math>, según la ecuación <math>Q(s,a) \leftarrow (1 - \alpha)Q(s,a) + \alpha [r_{s,a} + \gamma \max_{a'} Q(s',a')]</math></li> <li>▪ <math>s \leftarrow s'</math></li> <li>▪ Informar de que el agente ha acabado el ciclo.</li> </ul> </li> <li>o Hasta que <math>s</math> sea un estado final o se cumpla el límite de ciclos para un episodio</li> <li>o Informar de que el agente ha acabado el episodio.</li> <li>o Informar de las estadísticas del episodio completado.</li> </ul> </li> <li>- Hasta que se cumpla el límite de episodios previsto.</li> <li>- Informar de que el agente ha acabado el entrenamiento.</li> </ul>
--

**Ilustración 27 : Algoritmo de Q-learning adaptado para establecer las comunicaciones necesarias e interactuar con el agente de control en cada ciclo y en cada episodio**

### 3.1.3. Descripción de los estados y acciones

Los estados del proceso de aprendizaje serán de dos tipos:

- Cuando un agente disfrute del **uso exclusivo** del entorno, sólo será necesario que conozca sus **coordenadas** dentro del tablero; de esta forma, el estado viene descrito por

$$S_e(a) = (f_i, c_j) \quad i = 0, \dots, n^\circ \text{ filas del entorno} - 1, j = 0, \dots, n^\circ \text{ columnas del entorno} - 1.$$

- Cuando un agente haga **uso compartido** el entorno con otros agentes, que también pueden moverse y con los que puede chocar, además de las coordenadas en el tablero será necesario que conozca cuál es la **situación de las casillas adyacentes** (libres, ocupadas por un agente de un equipo determinado o por un obstáculo).

$$S_c(a) = (f_i, c_j, s_n, s_o, s_s, s_e) \\ i = 0, \dots, n^\circ \text{ filas del entorno} - 1 \\ j = 0, \dots, n^\circ \text{ columnas del entorno} - 1, \\ s_k = \text{libre, obstaculo, jugador equipo } x, k = \text{norte, oeste, sur, este}$$

En cuanto al conjunto de acciones en el entorno, sólo habrá **cinco acciones permitidas**:

- Desplazamiento desde casilla hacia el norte.
- Desplazamiento desde casilla hacia el oeste.
- Desplazamiento desde casilla hacia el sur.
- Desplazamiento desde casilla hacia el este.
- Mantener su posición.

### 3.1.4. Política de decisión de acción $\varepsilon$ – greedy variable

La política de decisión de acción será  $\varepsilon$  – **greedy variable**[20], siendo  $\varepsilon$  **configurable** por el usuario en el caso de las implementaciones del algoritmo que disfruten en exclusiva del entorno.

Para las implementaciones de algoritmos que compartan el entorno con otros agentes la política  $\varepsilon$  – **greedy** será modificada, de forma que pueda **evolucionar** durante el proceso de aprendizaje, para así lograr un mejor equilibrio entre la exploración del entorno y la explotación del conocimiento adquirido.

Dicha modificación viene dada por un **decremento / incremento del parámetro  $\varepsilon$**  que permitirá reducir la probabilidad de tomar acciones aleatorias cuando un agente logre llegar a su objetivos e incrementará dicha probabilidad cuando, en el límite de ciclos, el agente no

consiga llegar a su objetivo. Por ejemplo, en un hipotético proceso de 100 episodios en el que el agente siempre consigue llegar a su objetivo, empezando con  $\epsilon = 0.7$  y usando un decremento de 0.01, la evolución de las probabilidades de política de decisión de acción sería:

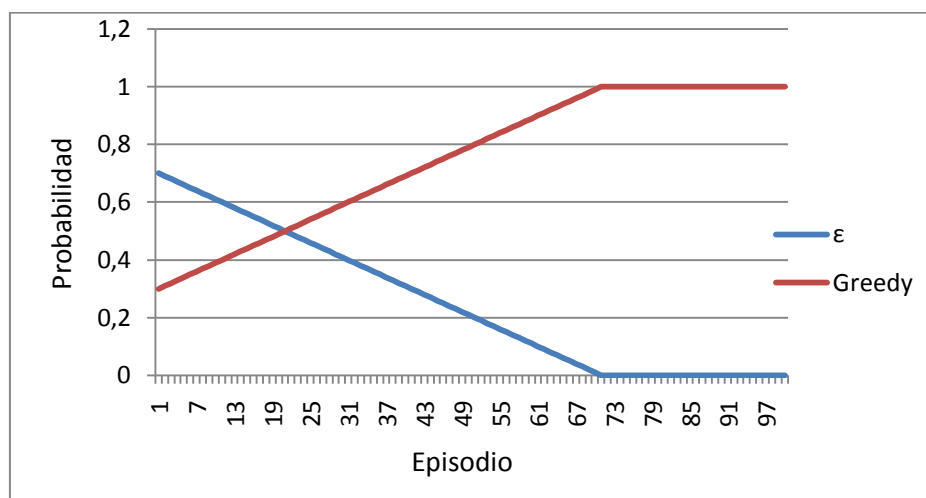


Ilustración 28 : Ejemplo de  $\epsilon$  - greedy con  $\epsilon$  variable, sólo incrementos

En una evolución en la que el agente no llegara siempre a su objetivo, tomando como incremento 0.1, la evolución de la política de decisión de acción sería:

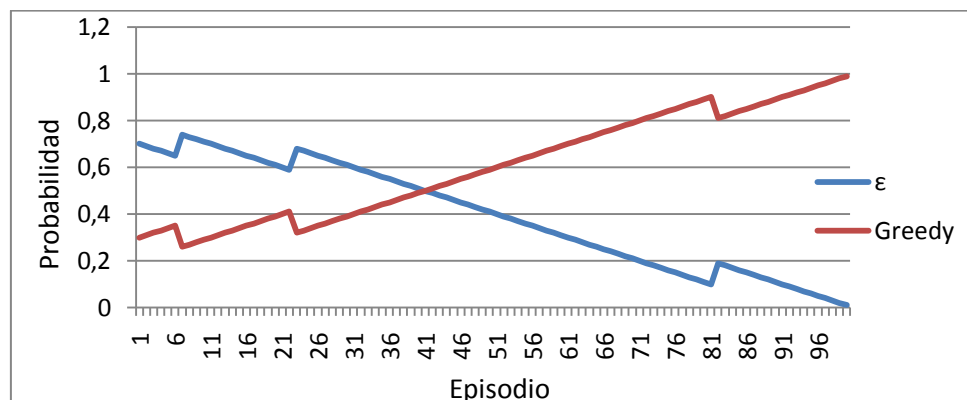


Ilustración 29 : Ejemplo de  $\epsilon$  - greedy con  $\epsilon$  variable, con incrementos y decrementos.

En este ejemplo, se ha supuesto que en los episodios 5, 21 y 81 el agente no consigue llegar a su objetivo, por lo que  $\epsilon$  se incrementa 0.1. En el resto de episodios el agente consigue llegar normalmente a su objetivo, por lo que  $\epsilon$  se decrementa en 0.01.

### 3.2. Organizaciones

El modelo de organizaciones está pensado para, cuando la complejidad del sistema crece, poder mantener los principios de modularidad y encapsulación que un sistema

informático requiere. En cada organización, un agente puede jugar uno o más roles, interactuando con otros agentes mediante protocolos para compartir conocimientos y coordinar sus actividades. El sistema estará compuesto por seis organizaciones que colaborarán entre sí para la obtención de la solución.

### **3.2.1. Organización Control**

El propósito de la organización Control es **organizar el acceso ordenado** a los recursos del sistema, garantizando el acceso exclusivo a ellos cuando sea necesario.

La organización Control mantiene información acerca del estado de todos los agentes implicados en algún proceso de aprendizaje. Dichos agentes tienen la obligación de actualizar dicha información tan pronto como sea modificada para que Control pueda dar paso a otros agentes al uso de los recursos compartidos.

### **3.2.2. Organización Tablero**

El propósito de la organización Tablero es gestionar el **acceso a los recursos del entorno Tablero**.

Será la encargada de gestionar la obtención y modificación del estado de los agentes mediante movimientos, reinicios o peticiones de estado. Los agentes implicados en algún proceso de aprendizaje deberán solicitar el uso de los recursos del Tablero a través de esta organización.

### **3.2.3. Organización Repositorio**

El propósito de la organización Repositorio es **almacenar y gestionar el conocimiento** adquirido durante los distintos procesos de aprendizaje.

Las organizaciones de agentes implicadas en procesos de aprendizaje deben informar del conocimiento adquirido a la organización Repositorio. Además, pueden hacer uso de esta organización durante su proceso de aprendizaje para la obtención de información adicional generada por otros agentes.

### **3.2.4. Organización Estadística**

El propósito de la organización Estadística es la **obtención de estadísticas** acerca de los diferentes procesos de aprendizaje.

Los agentes implicados en procesos de aprendizaje deben informar de las estadísticas de su proceso tan pronto como se vayan completando etapas de dicho aprendizaje. Con esta

información, la organización Estadística generará informes para el usuario acerca de la evolución de los diferentes procesos de aprendizaje.

### **3.2.5. Organización de Jugadores en Entorno Exclusivo**

El propósito de la organización de Jugadores en Entorno Exclusivo es **aprender una política de acciones** que conduzca a la meta en el Tablero.

Para ello los agentes de la organización de Jugadores en Entorno Exclusivo disfrutarán en **exclusiva** de todos los recursos del entorno, sin que sus acciones sean limitadas por otros agentes.

En esta organización los agentes están divididos en los equipos que el problema especifique, teniendo cada equipo unos objetivos diferentes y, por tanto, un proceso de aprendizaje diferente.

### **3.2.6. Organización de Jugadores en Entorno Compartido**

El propósito de la organización de Jugadores en Entorno Compartido es **aprender una política de acciones** que conduzca a la meta en el Tablero.

A diferencia de los agentes de la organización de Jugadores en Entorno Exclusivo, los agentes de la organización Jugadores Entorno Compartido no tienen en exclusiva el uso de los recursos del entorno, siendo **compartidos** con otros agentes de su misma organización.

En esta organización los agentes están divididos en equipos, teniendo cada equipo diferentes objetivos y, por tanto, un proceso de aprendizaje diferente.

## **3.3. Modelo del entorno**

Existen tres entornos a los que las diferentes organizaciones pueden acceder para lograr sus objetivos:

1. **Repositorio.** Contiene el conocimiento adquirido a través de los diferentes procesos de aprendizaje, al que las organizaciones de Jugadores en Entorno Exclusivo y de Jugadores en Entorno Compartido pueden acceder. La organización Repositorio se encarga de su gestión.
2. **Tablero.** Contiene la información acerca de todos los elementos del entorno en el que se mueven los diferentes agentes en sus procesos de aprendizaje. La organización Tablero se encarga de su gestión.
3. **Estadísticas.** Contiene información acerca del proceso de aprendizaje. Cuando los diferentes procesos de aprendizaje vayan siendo completados por las

organizaciones de Jugadores Individuales y de Jugadores Colectivos, enviarán información para rellenar dichas estadísticas. La organización Estadística se encarga de su gestión.

### 3.3.1. Repositorio

El entorno Repositorio almacena el conocimiento generado en todos los procesos de aprendizaje. Concretamente, los guarda en forma de **políticas**.

Cada política se define como una tupla  $\langle E \rangle \rightarrow \langle A, V \rangle$  tal que:

- $E$  es el conjunto de estados en el que un agente puede encontrarse.
- $A$  es el conjunto de acciones que un agente puede ejecutar.
- $V$  es la valoración de la tupla, con el fin de actualizar la tupla si cualquier proceso de aprendizaje generara un movimiento mejor que el almacenado.

Por ejemplo, para el siguiente entorno Tablero

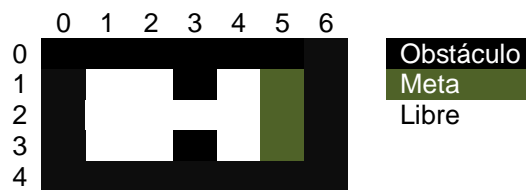


Ilustración 30 : Ejemplo de entorno Tablero

Una posible política obtenida al final de un proceso de aprendizaje (Q-Learning con  $\gamma = 0.95, \alpha = 1$ ) es

Estado	Acción	Valor
Fila 1 Columna 1	Movimiento hacia el sur	0.814506
Fila 2 Columna 1	Movimiento hacia el este	0.857375
Fila 3 Columna 1	Movimiento hacia el norte	0.814506
Fila 1 Columna 2	Movimiento hacia el sur	0.857375
Fila 2 Columna 2	Movimiento hacia el este	0.9025
Fila 3 Columna 2	Movimiento hacia el norte	0.857375
Fila 2 Columna 3	Movimiento hacia el este	0.95
Fila 1 Columna 4	Movimiento hacia el este	1
Fila 2 Columna 4	Movimiento hacia el este	1
Fila 3 Columna 4	Movimiento hacia el este	1

Ilustración 31 : Política obtenida mediante Q-Learning

Durante el proceso de aprendizaje se han podido obtener tuplas del tipo

$$\langle \text{Fila 3 Columna 4, Movimiento hacia el Norte} \rangle \rightarrow 0.95$$



obtenidas al recorrer caminos no óptimos y que habrían sido descartadas al conocer tuplas para un mismo estado y mayor valoración de los posibles movimientos.

Existe más de un tipo de política dentro del repositorio, debido a que en el sistema se dan dos tipos de aprendizaje, uno en entorno exclusivo y otro en entorno compartido. De esta forma, el entorno Repositorio guarda dos tipos de políticas:

- **Políticas obtenidas mediante aprendizaje en entorno exclusivo.** Reflejan el conocimiento obtenido mediante procesos de aprendizaje en entorno exclusivo en los que los agentes que la han desarrollado disfrutaban en exclusiva del entorno de movimiento. Dichas políticas reflejan el camino más rápido a la meta.
- **Políticas obtenidas mediante aprendizaje en entorno compartido.** Reflejan el conocimiento obtenido mediante procesos de aprendizaje en entorno compartido, en los que los agentes que la han desarrollado compartían el entorno de movimiento con otros agentes. Dichas políticas no tienen porqué reflejar el camino más rápido a la meta, puesto que los agentes que las generan pueden tener que cooperar con otros agentes.

Además, como existen agentes con diferentes objetivos en el entorno de movimiento, debe existir una política de cada tipo para cada objetivo. Los agentes que comparten un objetivo común están en un mismo equipo, por tanto, existe una política de movimientos individual para cada equipo y una política de movimientos colectivos para cada equipo.

De esta forma, el conjunto de acciones que se pueden efectuar sobre el entorno Repositorio consiste en:

Leer <i>Política individual</i> $[eq][e]$	$eq = 0, \text{número total de equipos}$
Escribir <i>Política individual</i> $[eq][e][a][v]$	$e = 0, \text{número total de estados}$
Leer <i>Política colectiva</i> $[eq]$	$a = 0, \text{número total de acciones}$
Escribir <i>Política colectiva</i> $[eq][e][a][v]$	$v = [-1, 1]$

**Ilustración 32 : Acciones asociadas al entorno Repositorio**

### 3.3.2. Tablero

El propósito del entorno Tablero es mantener la información acerca de todos los objetos que contiene el entorno en el que los agentes se mueven para lograr sus objetivos. El Tablero está formado por casillas que los agentes jugadores entienden como estados.

El Tablero guarda tres tipos de información:

- **Configuración** propia del entorno y del experimento.
- **Información estática**, información acerca de qué tipo es cada estado. Un estado puede ser de tipo obstáculo o no accesible, inicio, meta o libre.
- **Información dinámica**, acerca del estado en el que se encuentra cada agente jugador dentro del Tablero.

La configuración la componen una serie de parámetros que describen el experimento:

- Equipos que compiten en el entorno.
- Jugadores Entorno Compartido por equipo.
- Filas y columnas del entorno.
- Nombre del entorno.
- Descripción del entorno.

La información estática está organizada en forma de tuplas, existiendo una tupla para los estados de inicio y una segunda tupla para los estados meta, en la forma  $\langle E, Eq \rangle$ , tal que

- $E$  es el conjunto de estados posibles en el entorno.
- $Eq$  es el conjunto de equipos existentes en el sistema.

Debido a que existen en el sistema varios equipos con diferentes inicios y metas, un estado puede ser de diferentes tipos para diferentes equipos:

$$\langle E_x, Eq_i \rangle \leftrightarrow \langle E_x, Eq_j \rangle$$

Para los estados obstáculo o no accesibles, se almacena únicamente el estado del que se trata, puesto que son obstáculos para todos los equipos.

La información dinámica está organizada en forma de tupla, existiendo una primera tupla que almacena el estado de cada agente  $\langle A, E \rangle$ , tal que:

- $A$  es el conjunto de los agentes jugadores presentes en el Tablero.
- $E$  es el conjunto de estados posibles en el Tablero.

y una segunda tupla que almacena el equipo de cada agente,  $\langle A, Eq \rangle$ , tal que:

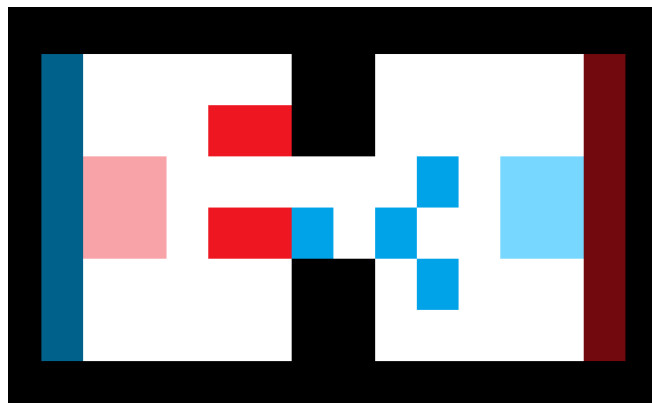
- $A$  es el conjunto de los agentes jugadores presentes en el Tablero.
- $Eq$  es el conjunto de equipos existentes en el sistema.

De esta forma, el conjunto de acciones que se pueden efectuar sobre el entorno Tablero son:

Leer <i>configuracion</i>	$eq = 0$ , número total de equipos
Leer <i>Inicio</i> $[i][eq]$	$i = 0$ , número total de inicios para el equipo $eq$
Leer <i>Meta</i> $[m][eq]$	$m = 0$ , número total de metas para el equipo $eq$
Leer <i>Obstaculo</i> $[e]$	$e = 0$ , número total de estados
Leer <i>Posicion</i> $[a]$	
Escribir <i>Posicion</i> $[a][e]$	
Borrar <i>Posicion</i> $[a][e]$	
Leer <i>Equipo</i> $[a]$	
Escribir <i>Equipo</i> $[a][eq]$	

**Ilustración 33 : Acciones asociadas al entorno Tablero**

En el siguiente ejemplo



**Ilustración 34 : Ejemplo de entorno Tablero**

La configuración del Tablero es:

- 2 equipos.
- 4 agentes Jugadores Entorno Compartido por equipo.
- 9 filas y 16 columnas.

La información estática almacenada por el Tablero sería:

Obstáculos		Inicios			Metas		
Estado		Estado		Equipo	Estado		Equipo
Fila	Columna	Fila	Columna		Fila	Columna	
0	0..15	3	2	0	1	14	0
1	0	3	3	0	2	14	0
1	7	4	2	0	3	14	0
1	8	4	3	0	4	14	0
1	15	3	12	1	5	14	0
2	0	3	13	1	6	14	0
2	7	4	12	1	1	1	1
2	8	4	13	1	2	1	1
2	15				3	1	1
5	0				4	1	1
5	7				5	1	1
5	8				6	1	1
5	15						
6	0						
6	7						
6	8						
6	15						
7	0..15						

Tabla 3 : Información estática para el entorno Tablero de la ilustración 27.

Y la información dinámica en el momento representado en la ilustración 27 sería:

Posición			Equipo	
Jugador	Estado		Jugador	Equipo
	Fila	Columna		
1	2	5	1	0
2	4	7	2	1
3	2	5	3	0
4	4	9	4	1
5	4	5	5	0
6	3	10	6	1
7	4	6	7	0
8	5	10	8	1

Tabla 4 : Información dinámica para el entorno Tablero de la ilustración 27

### 3.3.3. Estadísticas

El propósito del entorno Estadísticas es mantener la información acerca de cómo evoluciona el proceso de aprendizaje, las **estadísticas** que dicho aprendizaje genera. Debido a que existen en el sistema dos procesos de aprendizaje diferentes, el sistema guardará dos tipos de información:

- **Estadísticas del proceso de aprendizaje exclusivo**, generadas por los agentes de la organización de Jugadores en Entorno Exclusivo durante su proceso de aprendizaje.
- **Estadísticas del proceso de aprendizaje compartido**. Son las generadas por los agentes de la organización de Jugadores en Entorno Compartido durante su proceso de aprendizaje.

Las estadísticas del proceso de aprendizaje exclusivo están organizadas según la tupla  $\langle Ag, Ep \rangle \rightarrow \langle M_a, M_{na}, M_T, M_n, O \rangle$ , tal que

- $Ag$  es el agente perteneciente a la organización de Jugadores Entorno Exclusivo que ha efectuado el episodio de aprendizaje.
- $Ep$  es el número de orden del episodio que el agente ha ejecutado en su proceso de aprendizaje exclusivo.
- $M_a$  es el número de movimientos aleatorios realizados durante el episodio.
- $M_{na}$  es el número de movimientos no aleatorios realizados durante el episodio.
- $M_T$  es el número de movimientos total realizados durante el episodio.
- $M_n$  es el número de movimientos cuyo resultado fue no válido
- $O$  indica si la meta fue alcanzada o no al final del episodio de aprendizaje.

Las estadísticas del proceso de aprendizaje exclusivo están organizadas según la tupla  $\langle Ag, Ep \rangle \rightarrow \langle M_a, M_i, M_c, M_T, M_n, O \rangle$ , tal que

- $Ag$  es el agente perteneciente a la organización de Jugadores Entorno Compartido que ha efectuado el episodio de aprendizaje.
- $Ep$  es el número de orden del episodio que el agente ha ejecutado en su proceso de aprendizaje compartido.
- $M_a$  es el número de movimientos aleatorios realizados durante el episodio.
- $M_i$  es el número de movimientos según la política exclusiva realizados en el episodio.
- $M_c$  es el número de movimientos según la política compartida realizados durante el episodio.
- $M_T$  es el número de movimientos total que se tomaron durante el episodio.
- $M_n$  es el número de movimientos cuyo resultado no fue válido.
- $O$  indica si la meta fue alcanzada o no al final del episodio de aprendizaje compartido.

De esta forma, el conjunto de acciones que se pueden efectuar sobre el entorno Estadístico viene dado por:

Escribir	<i>EstadísticasExclusivas</i> [ $Ag_i$ ][ $Ep_j$ ] $i = 0$ , número total de jugadores entorno exclusivo $j = 0$ , número total de episodios en el proceso de aprendizaje exclusivo
Escribir	<i>EstadísticasCompartidas</i> [ $Ag_i$ ][ $Ep_j$ ] $i = 0$ , número total de jugadores entorno compartido $j = 0$ , número total de episodios en el proceso de aprendizaje compartido

**Ilustración 35 : Acciones asociadas al entorno Estadístico**

### 3.4. Modelo preliminar de roles

A continuación se exponen los roles necesarios para el sistema. Este modelo es incompleto, puesto que sólo describe los roles que no comprometen el diseño con una arquitectura concreta. La especificación según Gaia se encuentra en el anexo I, encontrándose a continuación una versión resumida de dicho modelo.

#### 3.4.1. Gestión del entorno Tablero

Para la gestión del entorno Tablero existirán tres roles que se encarguen de las tareas:

- Extraer el estado de un agente.
- Reiniciar la posición de un agente.
- Ejecutar una acción solicitada por un agente y devolverle el refuerzo asociado a dicha acción.

#### 3.4.2. Gestión del entorno Estadístico

Para la gestión del entorno Estadístico existirán dos roles que se encarguen de las tareas:

- Recoger las estadísticas enviadas por un agente.
- Procesar las estadísticas que se hayan recogido hasta el momento.

#### 3.4.3. Gestión del entorno Repositorio

Para la gestión del entorno Repositorio existirán tres roles que se encarguen de las tareas:

- Recoger el conocimiento enviado por un agente que disfruta en exclusiva del entorno Tablero.
- Enviar el conocimiento adquirido por agentes que usan en exclusiva el entorno Tablero a agentes que lo requieran.

- Recoger el conocimiento enviado por un agente que comparte el entorno Tablero con otros agentes.
- Enviar el conocimiento adquirido por agentes que comparten el entorno Tablero con otros agentes a los agentes que lo requieran.
- Procesar el conocimiento recibido, generando las políticas con dicho conocimiento.

Las tareas relativas a la gestión del conocimiento obtenido por agentes que disfrutan del entorno en exclusiva serán realizadas por un mismo rol. Así mismo, la gestión del conocimiento obtenido por agentes que comparten el entorno con otros agentes serán también realizadas por un mismo rol.

#### **3.4.4. *Jugadores en Entorno Exclusivo***

Para ejecutar el algoritmo de aprendizaje en un entorno exclusivo será necesario completar un único rol, el del agente que ejecutará procesos usando en exclusiva el entorno.

#### **3.4.5. *Jugadores en Entorno Compartido***

Para ejecutar el algoritmo de aprendizaje en un entorno compartido será necesario completar un único rol, el del agente que ejecutará procesos compartiendo el entorno.

#### **3.4.6. *Control de los procesos de aprendizaje***

Existirán dos procesos de aprendizaje diferentes que habrá que controlar de forma separada:

- Control del aprendizaje en entorno exclusivo. Esta tarea incluye garantizar que el agente que ejecuta el aprendizaje tiene el uso en exclusiva de todos los recursos del entorno Tablero durante todo su proceso de aprendizaje.
- Control del aprendizaje en entorno compartido. Esta tarea incluye garantizar que los agentes están en la misma fase del proceso de aprendizaje en el entorno Tablero que comparten.

De esta forma, existirá un rol para cada tipo de control necesario.

### **3.5. Modelo preliminar de interacciones**

A continuación se describen las interacciones básicas entre los roles anteriormente descritos. De nuevo, este modelo es incompleto, puesto que sólo describe las interacciones que no comprometen el diseño con una arquitectura concreta. Su descripción se encuentra en el anexo II.

### **3.5.1. Interacciones con la organización Tablero**

Existen tres posibles interacciones con la organización Tablero, suficientes para que los jugadores obtengan la información, reinicien su posición y ejecuten los movimientos necesarios para sus distintos procesos de aprendizaje:

### **3.5.2. Interacciones con la organización Estadística**

Sólo existe una interacción posible con la organización estadística, que es la de enviar las estadísticas generadas durante el proceso de aprendizaje correspondiente. La organización se encarga internamente del proceso de dichas estadísticas.

### **3.5.3. Interacciones con la organización Repositorio**

Las interacciones posibles con la organización Repositorio son las de solicitar y enviar el conocimiento adquirido en los diferentes procesos de aprendizaje. La organización repositorio se encarga internamente del proceso del conocimiento.

### **3.5.4. Interacciones con la organización de Jugadores en Entorno Exclusivo**

La única interacción posible con un agente Jugador en Entorno Exclusivo es aquella en la que se le da permiso para empezar su proceso de aprendizaje.

### **3.5.5. Interacciones con la organización de Jugadores en Entorno Compartido**

Las únicas interacciones posibles con un agente Jugador en Entorno Compartido son aquellas en las que se le da permiso para empezar alguna parte de su proceso de aprendizaje, ya sea un episodio o un ciclo.

### **3.5.6. Interacciones con la organización de control**

Las interacciones con la organización control son las que efectúan los agentes cuando están preparados para empezar o han acabado parte de sus respectivos procesos de aprendizaje.

## **3.6. Normas de la organización**

Las normas de la organización especifican que roles existen y que interacciones pueden ejecutar. Además especifica el orden en el que deben ser ejecutadas dichas interacciones. Para desarrollar las normas se ha seguido un proceso constructivo, en el que paso a paso se va construyendo una norma completa según los pasos necesarios en el algoritmo especificado en el apartado 3.1. La notación de las normas se especifica en el anexo VIII.



### 3.6.1. Normas de servicio

Del modelo preliminar de roles se extraen 12 roles diferentes:

- R1. Gestor de estado
- R2. Gestor de reinicio
- R3. Gestor de movimiento
- R4. Gestor estadístico
- R5. Procesador estadístico
- R6. Gestor de conocimiento individual
- R7. Gestor de conocimiento colectivo
- R8. Procesador de conocimiento
- R9. Jugador en entorno exclusivo
- R10. Jugador en entorno compartido
- R11. Control del entrenamiento en entorno exclusivo
- R12. Control del entrenamiento en entorno compartido

En cuanto a su multiplicidad, los roles control del entrenamiento en entorno exclusivo y control del entrenamiento en entorno compartido han de ser únicos en el sistema.

$$R_i^{1...1}, \quad i = 11, 12$$

Los roles Gestor de estado, Gestor de reinicio, Gestor de movimiento, Gestor estadístico, Procesador estadístico, Gestor de conocimiento individual, Gestor de conocimiento colectivo y Procesador de conocimiento pueden ser representados por varias entidades al mismo tiempo.

$$R_i^{1...N}, \quad i = 1, 2, 3, 4, 5, 6, 7, 8$$

el rol jugador en entorno exclusivo ha de ser desempeñado por un único agente por equipo,

$$R_9^{j...j}, \quad j = \text{número de equipos}$$

y, por último, el rol jugador en entorno compartido ha de ser desempeñado por el mismo número de agentes por cada equipo,

$$R_{10}^{j*k...j*k}, \quad j = \text{número de equipos}, \quad k = \text{miembros colectivos por equipo}$$

En cuanto a la exclusividad, los roles gestores, de proceso y de control no pueden ser desempeñados por ningún tipo de jugador.

$$\neg (R_i \mid R_j), \quad i = 1, 2, 3, 4, 5, 6, 7, 8, 11, 12 \quad j = 9, 10$$

### 3.6.2. Normas de las interacciones

De igual forma, del modelo preliminar de interacciones podemos extraer 18 posibles interacciones o protocolos:

- P1. Pedir estado
- P2. Pedir reinicio
- P3. Pedir movimiento
- P4. Enviar estadísticas
- P5. Enviar conocimiento
- P6. Pedir conocimiento exclusivo
- P7. Pedir conocimiento compartido
- P8. Empezar entrenamiento en entorno exclusivo
- P9. Empezar episodio de entrenamiento en entorno compartido
- P10. Empezar ciclo de entrenamiento en entorno compartido
- P11. Listo para entrenamiento en entorno exclusivo
- P12. Entrenamiento en entorno exclusivo acabado
- P13. Listo para episodio de entrenamiento en entorno compartido
- P14. Episodio de entrenamiento en entorno compartido acabado
- P15. Listo para ciclo de entrenamiento en entorno compartido
- P16. Ciclo de entrenamiento en entorno compartido acabado
- P17. Entrenamiento en entorno compartido acabado
- P18. Pedir registro en entorno Tablero

### 3.6.3. Exclusividad de las interacciones

Muchas de estas interacciones sólo pueden ser ejecutadas por un rol. Concretamente, sólo Control del entrenamiento en entorno exclusivo puede ejecutar la interacción

- P8. Empezar entrenamiento en entorno exclusivo

$$P_8(R_{11})$$

Sólo Control del entrenamiento en entorno compartido puede ejecutar las interacciones

- P9. Empezar episodio de entrenamiento en entorno compartido
- P10. Empezar ciclo de entrenamiento en entorno compartido

$$P_i(R_{12}), \quad i = 9, 10$$

De igual forma, sólo el rol jugador en entorno exclusivo puede ejecutar las interacciones

P11. Listo para entrenamiento en entorno exclusivo

P12. Entrenamiento en entorno exclusivo acabado

$$P_i(R_9), \quad i = 11, 12$$

Y, sólo el rol jugador en entorno compartido puede ejecutar las interacciones

P6. Pedir conocimiento exclusivo

P7. Pedir conocimiento compartido

P13. Listo para episodio de entrenamiento en entorno compartido

P14. Episodio de entrenamiento en entorno compartido acabado

P15. Listo para ciclo de entrenamiento en entorno compartido

P16. Ciclo de entrenamiento en entorno compartido acabado

P17. Entrenamiento en entorno compartido acabado

$$P_i(R_{10}), \quad i = 6, 7, 13, 14, 15, 16, 17$$

El rol jugador en entorno exclusivo y el rol jugador en entorno compartido son los únicos que pueden ejecutar las interacciones

P1. Pedir estado

P2. Pedir reinicio

P3. Pedir movimiento

P4. Enviar estadísticas

P5. Enviar conocimiento

P18. Pedir registro en entorno Tablero

$$P_i(R_j), \quad i = 1, 2, 3, 4, 5, \quad j = 9, 10$$

#### **3.6.4. Autorizaciones genéricas para los procesos de aprendizaje**

Ningún agente jugador puede usar servicios de los gestores de cualquier entorno sin haber pedido su registro en el entorno Tablero previamente.

$$P_{18}(R_j) \rightarrow P_i(R_j), \quad i = 1, 2, 3, 4, 5, 6, 7, \quad j = 9, 10$$

#### **3.6.5. Autorizaciones en el proceso de aprendizaje en entorno exclusivo**

Ningún jugador será autorizado para empezar su entrenamiento si antes no ha solicitado permiso para ello. Concretamente, el jugador en entorno exclusivo no será autorizado a empezar su proceso de aprendizaje si no pide permiso para ello primero

$$P_{11}(R_9) \rightarrow P_8(R_{11})$$

y, además, sólo puede pedir permiso una vez y una vez obtenido ejecutará un único proceso de aprendizaje, por lo que

$$P_{11}(R_9)^1 \rightarrow P_8(R_{11})^1$$

Finalmente, debe informar de que ha acabado el proceso de aprendizaje, por lo que

$$P_{11}(R_9)^1 \rightarrow P_8(R_{11})^1 \rightarrow P_{12}(R_9)^1$$

describe los protocolos de autorización que el rol jugador en entorno exclusivo debe seguir para ejecutar su proceso de aprendizaje.

### 3.6.6. Normas del proceso de aprendizaje en entorno exclusivo

El proceso de aprendizaje en entorno exclusivo empieza cuando un agente que desempeña el rol jugador en entorno exclusivo recibe la autorización para empezar su proceso de aprendizaje. En ese momento, lo primero que hace es pedir el reinicio de su posición en el entorno e informarse de la posición en la que ha sido iniciado.

$$P_{12}(R_9) \rightarrow P_2(R_9) \rightarrow P_1(R_9)$$

Una vez reiniciada y conocida su posición el jugador pide un movimiento y vuelve a informarse de cuál es su nuevo estado, es decir

$$P_{12}(R_9) \rightarrow P_2(R_9) \rightarrow P_1(R_9) \rightarrow P_3(R_9) \rightarrow P_1(R_9)$$

Pedirá tantos movimientos como indique la configuración de su proceso de aprendizaje, y después de cada movimiento informará de su nuevo estado.

$$P_{12}(R_9) \rightarrow P_2(R_9) \rightarrow P_1(R_9) \rightarrow ( \underbrace{P_3(R_9) \rightarrow P_1(R_9)}_{\text{Ciclo de aprendizaje individual}} )^+$$

Durante el proceso de aprendizaje, el jugador individual volverá a reiniciar su posición al inicio de cada episodio, por lo que

$$\underbrace{P_{12}(R_9) \rightarrow (P_2(R_9) \rightarrow P_1(R_9) \rightarrow ( \underbrace{P_3(R_9) \rightarrow P_1(R_9)}_{\text{Ciclo de aprendizaje individual}} )^+ )^+}_{\text{Episodio de aprendizaje individual}}$$

Una vez acabado el proceso de aprendizaje en entorno exclusivo, el jugador informará de las estadísticas obtenidas y del conocimiento adquirido.

$$\underbrace{P_{12}(R_9) \rightarrow (P_2(R_9) \rightarrow P_1(R_9) \rightarrow ( \underbrace{P_3(R_9) \rightarrow P_1(R_9)}_{\text{Ciclo de aprendizaje individual}} )^+ )^+}_{\text{Episodio de aprendizaje individual}} \rightarrow P_4(R_9) \rightarrow P_5(R_9)$$

Y por último, informará de la finalización del proceso de entrenamiento

$$P_{12}(R_9) \rightarrow (P_2(R_9) \rightarrow P_1(R_9) \rightarrow ( \underbrace{P_3(R_9) \rightarrow P_1(R_9)}_{\text{Ciclo de aprendizaje individual}} )^+ \rightarrow P_4(R_9) \rightarrow P_5(R_9) \rightarrow \textcolor{red}{P_{12}(R_9)})$$

*Episodio de aprendizaje individual*

### 3.6.7. Autorizaciones en el proceso de aprendizaje en entorno compartido

Análogamente a las autorizaciones en el proceso de aprendizaje en entorno exclusivo, un jugador en entorno compartido no será autorizado a empezar un episodio o un ciclo de su proceso de aprendizaje si no pide permiso para ello primero

$$P_{13}(R_{10}) \rightarrow P_9(R_{12}), \quad \text{para los episodios}$$

$$P_{15}(R_{10}) \rightarrow P_{10}(R_{12}), \quad \text{para los ciclos}$$

El proceso es único, es decir, se concederá una autorización por cada solicitud pedida, y no se podrá tener pendiente más de una autorización al mismo tiempo.

$$P_{13}(R_{10})^1 \rightarrow P_9(R_{12})^1, \quad \text{para los episodios}$$

$$P_{15}(R_{10})^1 \rightarrow P_{10}(R_{12})^1, \quad \text{para los ciclos}$$

y, además, no puede acabar un episodio o ciclo antes de ser autorizado, por lo que

$$P_{13}(R_{10})^1 \rightarrow P_9(R_{12})^1 \rightarrow \textcolor{red}{P_{14}(R_{10})^1}, \quad \text{para los episodios}$$

$$P_{15}(R_{10})^1 \rightarrow P_{10}(R_{12})^1 \rightarrow \textcolor{red}{P_{16}(R_{10})^1}, \quad \text{para los ciclos}$$

Para completar la relación entre episodios y ciclos, un episodio contiene varios ciclos. Los ciclos empezarán a ejecutarse una vez recibida la autorización para empezar el episodio y serán tantos como indique el proceso de aprendizaje, de forma que

$$P_{13}(R_{10})^1 \rightarrow P_9(R_{12})^1 \rightarrow \underbrace{(P_{15}(R_{10})^1 \rightarrow P_{10}(R_{12})^1 \rightarrow P_{16}(R_{10})^1)^+}_{\text{Ciclo de aprendizaje colectivo}} \rightarrow P_{14}(R_{10})^1$$

*Episodio de aprendizaje colectivo*

es la descripción de los protocolos de autorizaciones que el rol jugador en entorno compartido ha de seguir en cada episodio de su proceso de aprendizaje, por lo que el proceso completo queda en la forma

$$\textbf{3.6.8. } (P_{13}(R_{10})^1 \rightarrow P_9(R_{12})^1 \rightarrow \underbrace{(P_{15}(R_{10})^1 \rightarrow P_{10}(R_{12})^1 \rightarrow P_{16}(R_{10})^1)^+}_{\text{Ciclo de aprendizaje colectivo}} \rightarrow P_{14}(R_{10})^1) +$$

*Episodio de aprendizaje colectivo*

### 3.6.9. Normas del proceso de aprendizaje en entorno compartido

El proceso de aprendizaje en entorno compartido es análogo al proceso de aprendizaje en entorno exclusivo, salvando las autorizaciones necesarias para ejecutar las diferentes

partes del proceso. Un episodio se inicia cuando el jugador en entorno compartido recibe permiso de Control para empezar dicho episodio que será único.

$$P_9(R_{12})^1$$

Una vez recibido el permiso, el jugador inicia su episodio reiniciando su estado e informándose acerca de en qué estado ha sido iniciado

$$P_9(R_{12})^1 \rightarrow P_2(R_{10})^1 \rightarrow P_1(R_{10})^1$$

A continuación, el jugador Control empezar un ciclo de aprendizaje, pero para ello, primero ha de pedir la autorización de control y esperar dicha autorización

$$P_9(R_{12})^1 \rightarrow P_2(R_{10})^1 \rightarrow P_1(R_{10})^1 \rightarrow P_{15}(R_{10})^1 \rightarrow P_{10}(R_{12})^1$$

Una vez obtiene la autorización, empieza a ejecutar su ciclo de aprendizaje en entorno compartido, moviéndose en primer lugar e informándose de su estado en segundo lugar.

$$P_9(R_{12})^1 \rightarrow P_2(R_{10})^1 \rightarrow P_1(R_{10})^1 \rightarrow P_{15}(R_{10})^1 \rightarrow P_{10}(R_{12})^1 \rightarrow P_3(R_{10})^1 \rightarrow P_1(R_{10})^1$$

La decisión del movimiento a ejecutar puede requerir preguntar al Gestor de Conocimiento por la mejor acción obtenida en los procesos de aprendizaje en entorno exclusivo, por lo que

$$P_9(R_{12})^1 \rightarrow P_2(R_{10})^1 \rightarrow P_1(R_{10})^1 \rightarrow P_{15}(R_{10})^1 \rightarrow P_{10}(R_{12})^1 \rightarrow [P_6(R_{10})^1] \rightarrow P_3(R_{10})^1 \rightarrow P_1(R_{10})^1$$

Cuando el jugador conoce su estado ha acabado su ciclo de aprendizaje, por lo que informa de ello a Control.

$$P_9(R_{12})^1 \rightarrow P_2(R_{10})^1 \rightarrow P_1(R_{10})^1 \rightarrow P_{15}(R_{10})^1 \rightarrow P_{10}(R_{12})^1 \rightarrow [P_6(R_{10})^1] \rightarrow P_3(R_{10})^1 \rightarrow P_1(R_{10})^1 \rightarrow P_{16}(R_{12})^1$$

El jugador en entorno compartido volverá a pedir permiso tantas veces como ciclos necesite ejecutar en su episodio del proceso de aprendizaje

$$P_9(R_{12})^1 \rightarrow P_2(R_{10})^1 \rightarrow P_1(R_{10})^1 \rightarrow \underbrace{(P_{15}(R_{10})^1 \rightarrow P_{10}(R_{12})^1 \rightarrow [P_6(R_{10})^1] \rightarrow P_3(R_{10})^1 \rightarrow P_1(R_{10})^1 \rightarrow P_{16}(R_{12})^1)^+}_{\text{Ciclo de aprendizaje colectivo}}$$

Antes de terminar el episodio, el agente jugador colectivo informa de las estadísticas obtenidas al Gestor Estadístico y del conocimiento adquirido al Gestor de Conocimiento

$$P_9(R_{12})^1 \rightarrow P_2(R_{10})^1 \rightarrow P_1(R_{10})^1 \rightarrow (\underbrace{\dots \dots \dots}_{\text{Ciclo de aprendizaje colectivo}})^+ \rightarrow P_4(R_{10})^1 \rightarrow P_5(R_{10})^1$$

Es posible que al acabar el episodio, el jugador colectivo quiera intercambiar la información adquirida mediante el gestor de conocimiento

$$P_9(R_{12})^1 \rightarrow P_2(R_{10})^1 \rightarrow P_1(R_{10})^1 \rightarrow ( \underbrace{\dots\dots\dots}_{\text{Ciclo de aprendizaje colectivo}} )^+ \rightarrow P_4(R_{10})^1 \rightarrow P_5(R_{10})^1 \rightarrow [P_7(R_{10})^1]$$

Una vez acabado el episodio de aprendizaje en entorno compartido, el jugador colectivo informa de dicha situación a Control

$$P_9(R_{12})^1 \rightarrow P_2(R_{10})^1 \rightarrow P_1(R_{10})^1 \rightarrow ( \underbrace{\dots\dots\dots}_{\text{Ciclo de aprendizaje colectivo}} )^+ \rightarrow P_4(R_{10})^1 \rightarrow P_5(R_{10})^1 \rightarrow [P_7(R_{10})^1] \rightarrow P_{14}(R_{10})^1$$

Por último, puede ejecutar tantos episodios de aprendizaje en entorno compartido como necesite, por lo que el proceso de aprendizaje en entorno compartido queda

$$P_9(R_{12})^1 \rightarrow (P_2(R_{10})^1 \rightarrow P_1(R_{10})^1 \rightarrow ( \underbrace{\dots\dots\dots}_{\text{Ciclo de aprendizaje colectivo}} )^+ \rightarrow P_4(R_{10})^1 \rightarrow P_5(R_{10})^1 \rightarrow [P_7(R_{10})^1] \rightarrow P_{14}(R_{10})^1)^+$$

*Episodio de aprendizaje colectivo*

Siendo el ciclo de aprendizaje en entorno compartido

$$\mathbf{3.6.10.} \quad \underbrace{P_{15}(R_{10})^1 \rightarrow P_{10}(R_{12})^1 \rightarrow [P_6(R_{10})^1] \rightarrow P_3(R_{10})^1 \rightarrow P_1(R_{10})^1 \rightarrow P_{16}(R_{12})^1}_{\text{Ciclo de aprendizaje colectivo}}$$

### 3.6.11. Normas para el inicio del entrenamiento en entorno exclusivo

Para empezar el entrenamiento en entorno exclusivo, Control del entrenamiento en entorno exclusivo debe recibir peticiones de inicio de procesos en entorno exclusivo por parte de los jugadores en entorno exclusivo

$$P_{11}(R_9)$$

Concretamente, debe recibir una por cada equipo presente en el entorno

$$(P_{11}(R_9))^i, \quad i = 0, \text{número de equipos}$$

Cuando tenga todas las peticiones, empezará el proceso de aprendizaje en entorno exclusivo de uno de ellos, autorizándole para comenzar.

$$(P_{11}(R_9))^i \rightarrow P_8(R_{11})$$

Y esperará a ser informado de la finalización del proceso de aprendizaje

$$(P_{11}(R_9))^i \rightarrow P_8(R_{11}) \rightarrow P_{12}(R_9)$$

Cuando acaba el proceso de aprendizaje en entorno exclusivo da paso al siguiente y así, uno por uno, van ejecutando su proceso todos los jugadores en entorno exclusivo, quedando el control del proceso de aprendizaje en entorno exclusivo en la forma

$$(P_{11}(R_9))^i \rightarrow (P_8(R_{11}) \rightarrow P_{12}(R_9))^i, \quad i = 0, \text{número de equipos}$$

### 3.6.12. Normas para el inicio de un episodio de aprendizaje en entorno compartido

Para empezar un episodio de entrenamiento en entorno compartido, Control debe recibir peticiones para empezar un episodio de aprendizaje en entorno compartido por parte de los jugadores en entorno compartido.

$$P_{11}(R_9)$$

Debe recibir tantas como jugadores en entorno compartido son esperados en el entorno

$$(P_{11}(R_9))^{i*j}, \quad i = 0, \text{número de equipos}, j = 0, \text{jugadores en entorno compartido por equipo}$$

Cuando tiene las peticiones que espera empieza el episodio de entrenamiento en entorno compartido de todos los jugadores al mismo tiempo, mediante la pertinente autorización.

$$(P_{11}(R_9))^{i*j} \rightarrow (P_9(R_{12}))^{i*j}$$

Es posible que algunos jugadores en entorno compartido hayan acabado su proceso de aprendizaje en entorno compartido y, por tanto, no necesiten ejecutar nuevos episodios, por lo que no habrá que contar con ellos para los posteriores episodios de aprendizaje. Así, el control del proceso de aprendizaje en entorno compartido queda en la forma

$$(P_{11}(R_9))^{(i*j)-k} \rightarrow (P_9(R_{12}))^{(i*j)-k}$$

$$i = 0, \text{número de equipos}, j = 0, \text{jugadores en entorno compartido por equipo}$$

$$k = \text{jugadores que han acabado su proceso de aprendizaje en entorno compartido}$$

### 3.6.13. Normas para el inicio de un ciclo de aprendizaje en entorno compartido

Para empezar un ciclo de entrenamiento en entorno compartido, Control debe recibir peticiones para empezar un ciclo de aprendizaje en entorno compartido por parte de los jugadores en entorno compartido.

$$P_{15}(R_9)$$

Concretamente debe recibir tantas como jugadores en entorno compartido son esperados en el entorno

$$(P_{15}(R_9))^{i*j}, \quad i = 0, \text{número de equipos}, j = 0, \text{jugadores en entorno compartido por equipo}$$

Cuando tiene las peticiones que espera, empieza el ciclo de entrenamiento en entorno compartido uno de ellos mediante la pertinente autorización.



$$(P_{15}(R_9))^{i*j} \rightarrow P_9(R_{12})$$

Y espera a que el jugador autorizado acabe su ciclo de aprendizaje en entorno compartido

$$(P_{15}(R_9))^{i*j} \rightarrow P_9(R_{12}) \rightarrow P_{16}(R_9)$$

Cuando el jugador autorizado acaba su ciclo de aprendizaje empieza el de otro nuevo, hasta completar el total de jugadores que han pedido autorización

$$(P_{15}(R_9))^{i*j} \rightarrow (P_9(R_{12}) \rightarrow P_{16}(R_9))^{i*j}$$

Es posible que algunos jugadores en entorno compartido hayan acabado su episodio de aprendizaje en entorno compartido y que, por tanto, no necesiten ejecutar más ciclos, por lo que no habrá que contar con ellos para los ciclos de aprendizaje. De esta forma, el control del ciclo de aprendizaje colectivo queda en la forma

$$(P_{15}(R_9))^{(i*j)-k} \rightarrow (P_9(R_{12}) \rightarrow P_{16}(R_9))^{(i*j)-k}$$

$i = 0$ , número de equipos,  $j = 0$ , jugadores en entorno compartido por equipo

$k =$  jugadores que han acabado su episodio de aprendizaje en entorno compartido

## 4. Diseño arquitectónico

En este capítulo se extiende el anterior, diseñando la arquitectura del sistema y sus organizaciones, explicando la integración del algoritmo de aprendizaje en dicha arquitectura y completando los modelos de roles e interacciones propuestos. Este capítulo se corresponde al Diseño Arquitectónico propuesto por la metodología Gaia, nuevamente extendiendo dicha metodología para describir la integración del algoritmo de aprendizaje en la arquitectura.

### 4.1. Representación de la estructura organizativa

Las seis organizaciones se estructuran según el siguiente gráfico

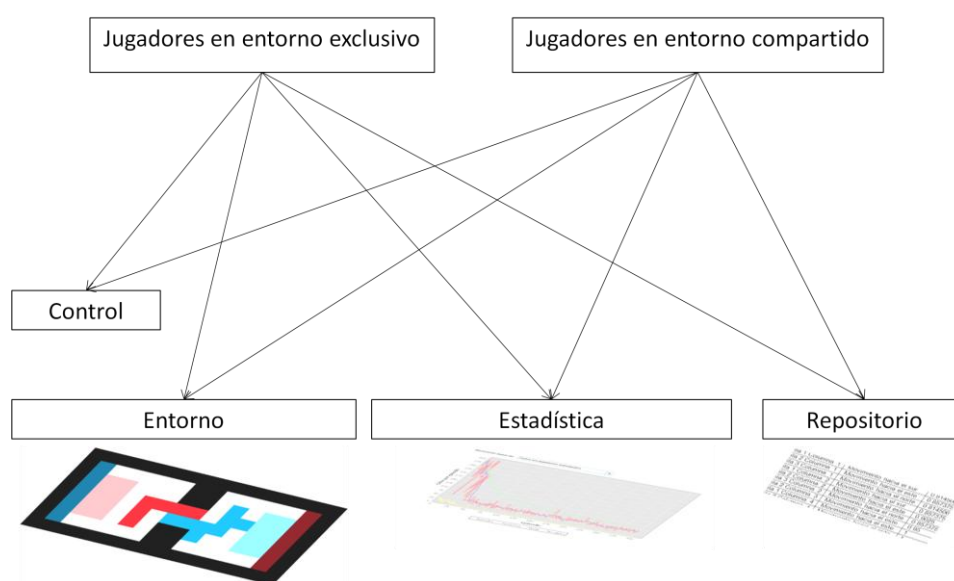


Ilustración 36 : Organizaciones presentes en el sistema

#### 4.1.1. Organización de control

La organización de control es la encargada de regular el uso del entorno Tablero por parte de las organizaciones de jugadores en entorno exclusivo y de jugadores en entorno compartido. Regula la ejecución de

- **Organización de jugadores en entorno exclusivo.** Se encarga de garantizar el acceso exclusivo durante el proceso de aprendizaje de los agentes jugador en entorno exclusivo.
- **Organización de jugadores en entorno compartido.** Se encarga de garantizar que todos los agentes jugador en entorno compartido se encuentran ejecutando a la vez el mismo episodio de aprendizaje y de garantizar el acceso exclusivo al entorno durante cada ciclo de aprendizaje.

La organización control contiene un único agente que cumple el rol Control del entrenamiento en entorno exclusivo y un único agente que cumple el rol Control del entrenamiento en entorno compartido.

#### **4.1.2. Organización Tablero**

La organización Tablero es la encargada de regular el acceso al entorno Tablero, en el que los agentes jugadores ejecutan sus procesos de aprendizaje.

Ejecuta las acciones de

- **Organización de jugadores en entorno exclusivo.** Se encarga de ejecutar las acciones, informar del estado y de reiniciar el estado de todos aquellos agentes que cumplen el rol jugador en entorno exclusivo.
- **Organización de jugadores en entorno compartido.** Se encarga de ejecutar las acciones, informar del estado y de reiniciar el estado de todos aquellos agentes que cumplen el rol jugador en entorno compartido.

La organización Tablero estará compuesta por agentes que pueden implementar uno o más de los roles Gestor Estado, Gestor Movimiento y Gestor Reinicio.

#### **4.1.3. Organización Repositorio**

La organización Repositorio es la encargada de almacenar, procesar y mostrar el conocimiento que adquieren los agentes mediante los diferentes procesos de aprendizaje.

Recibe conocimiento de

- **Organización de jugadores en entorno exclusivo.** Cada agente al acabar su proceso de aprendizaje en entorno exclusivo envía la política generada a la organización Repositorio.
- **Organización de jugadores en entorno compartido.** Cada agente al acabar un episodio de aprendizaje en entorno compartido envía el conocimiento adquirido acerca del episodio a la organización Repositorio.

Sirve conocimiento a la organización de jugadores en entorno compartido de dos formas:

- En todo momento, un jugador en entorno compartido puede pedir ser informado acerca de la mejor acción a ejecutar según la política en entorno exclusivo en un estado concreto.

- Al final de un episodio de aprendizaje en entorno exclusivo, un jugador en entorno exclusivo puede solicitar la política generada para su equipo por toda la organización de jugadores en entorno exclusivo.

La organización Repositorio estará compuesta por agentes encargados de recopilar el conocimiento y de agentes encargados de procesarla.

#### **4.1.4. Organización Estadística**

La organización Estadística es la encargada de almacenar, transformar y mostrar las estadísticas que generan los diferentes procesos de aprendizaje.

Recibe información de

- **Organización de jugadores en entorno exclusivo.** Cada agente al acabar un episodio de aprendizaje en entorno exclusivo envía la información acerca del episodio a la organización Estadística.
- **Organización de jugadores en entorno compartido.** Cada agente al acabar un episodio de aprendizaje en entorno compartido envía la información acerca del episodio a la organización Estadística.

La organización Estadística estará compuesta por agentes encargados de recopilar la información y de agentes encargados de procesarla y mostrarlos por pantalla.

#### **4.1.5. Organización de Jugadores en Entorno Exclusivo**

La organización de jugadores en entorno exclusivo es la organización encargada de adquirir el conocimiento sobre cómo alcanzar la solución en un entorno exclusivo, es decir, no tendrán que esquivar otros obstáculos aparte de los propios del entorno.

Depende de las organizaciones:

- **Organización de Control.** Se encarga de regular el acceso de los distintos miembros de la organización de jugadores en entorno exclusivo para garantizarles acceso exclusivo que necesitan en todo su proceso de aprendizaje.
- **Organización Entorno.** Se encarga de ejecutar los movimientos sobre el Entorno que los jugadores requieren para desarrollar su proceso de aprendizaje.
- **Organización Estadística.** Se encarga de recibir y almacenar la información que los miembros de la organización de jugadores en entorno exclusivo generan al finalizar cada episodio de su proceso de aprendizaje.

- **Organización Repositorio.** Se encarga de recibir y almacenar el conocimiento adquirido por los miembros de la organización de jugadores en entorno exclusivo.

La organización de jugadores en entorno exclusivo contiene múltiples agentes que cumplen el rol Jugador en entorno exclusivo, concretamente uno por cada equipo que intervenga en los procesos de aprendizaje.

#### **4.1.6. Organización de Jugadores en entorno compartido**

La organización de jugadores en entorno compartido es la encargada de adquirir el conocimiento sobre cómo alcanzar la solución en un entorno compartido, es decir, tendrán que tratar con otros jugadores que compiten y cooperar con ellos en la búsqueda de la solución.

Depende de las organizaciones:

- **Organización de Control.** Se encarga de regular el acceso de los distintos miembros de la organización de jugadores en entorno compartido para garantizarles acceso exclusivo que necesitan en cada ciclo de su proceso de aprendizaje y el acceso compartido que necesitan en los episodios de su proceso de aprendizaje.
- **Organización Entorno.** Se encarga de ejecutar los movimientos sobre el Entorno que los jugadores requieren para desarrollar su proceso de aprendizaje.
- **Organización Estadística.** Se encarga de recibir y almacenar la información que los miembros de la organización de jugadores en entorno compartido generan al finalizar cada episodio de su proceso de aprendizaje.
- **Organización Repositorio.** Se encarga de recibir y almacenar el conocimiento adquirido por los miembros de la organización de jugadores en entorno compartido. Además, informa a los miembros de la organización de jugadores en entorno compartido del mejor movimiento que la organización de jugadores en entorno exclusivo ha obtenido para un estado determinado y de la política que están desarrollando conjuntamente todos los miembros de la organización de jugadores en entorno compartido.

La organización de jugadores en entorno compartido contiene múltiples agentes que cumplen el rol Jugador en entorno compartido. El entorno determina cuántos puede haber como máximo, y ha de haber igual número de jugadores por cada equipo implicado en el proceso.

Además, cada uno de ellos cumplirá también el rol Buscador de Agentes, puesto que han de buscar a los agentes que necesitan para su ejecución.

## 4.2. Integración del algoritmo de aprendizaje en la arquitectura

En la arquitectura estarán presentes **dos procesos diferentes de aprendizaje**. En primer lugar, la organización de jugadores en entorno exclusivo ejecutará su **proceso de entrenamiento en entorno exclusivo**. En dicho entrenamiento interviene un jugador por cada equipo con el objetivo de obtener una política de movimientos sobre un entorno Tablero del que se disfruta en exclusiva. De esta forma, al acabar todos los procesos de aprendizaje, los agentes Jugador en entorno exclusivo habrán transferido al entorno Repositorio el conocimiento, en forma de políticas, que posteriormente usarán los agentes Jugadores en entorno compartido.

Una vez acabado el proceso de entrenamiento en entorno exclusivo, empieza el **proceso de entrenamiento en entorno compartido**, en el cual los agentes de la organización de Jugadores en entorno compartido intentarán aprender las políticas de movimientos que les lleven a la meta de la forma más rápida posible en un entorno Tablero compartido con otros agentes.

El hecho de que el entorno Tablero sea compartido es especialmente significativo, puesto que los agentes han de cooperar entre ellos para encontrar una buena política de movimientos. La cooperación será indirecta, a través del entorno Repositorio, en el cual los agentes depositarán el conocimiento adquirido y tomarán el adquirido por otros.

Además, los agentes Jugadores en entorno compartido pueden contar con el conocimiento adquirido previamente por los agentes Jugadores en entorno exclusivo, como guía de su proceso de aprendizaje.

## 4.3. Modelo de roles

El propósito del modelo de roles es completar el anterior modelo preliminar de roles, en el que se exponían los roles necesarios para el sistema sin comprometerse con una arquitectura en concreto.

El modelo de roles completo especifica todos los roles necesarios para la implementación del sistema multiagente. Se encuentra detallado en el anexo I, a continuación se explican los cambios realizados frente al modelo preliminar de roles.

### 4.3.1. Gestión del entorno Estadístico

Las estadísticas será recogidas por dos roles diferentes, según el tipo de estadística de la que se trate:

- Recoger las estadísticas enviadas por un jugador en entorno exclusivo.
- Recoger las estadísticas enviadas por un jugador en entorno compartido.

#### **4.3.2. Control de los procesos de aprendizaje**

Existen dos procesos de aprendizaje diferentes que han de ser controlados por separado.

En primer lugar el **aprendizaje en entorno exclusivo**. Esta tarea incluye garantizar que el agente que ejecuta el aprendizaje tiene el uso en exclusiva de todos los recursos del entorno Tablero durante todo su proceso de aprendizaje. Para ello, el rol Control del entrenamiento en entorno exclusivo esperará tener tantas peticiones de entrenamiento exclusivo como agentes. A continuación dará paso a todos los agentes, de forma aleatoria, esperando a que el primero acabe para dar paso al segundo y así sucesivamente.

En segundo lugar el **aprendizaje en entorno compartido**. Esta tarea incluye garantizar que los agentes están en la misma fase del proceso de aprendizaje en el entorno Tablero que comparten. Cada agente ejecuta un número determinado de episodios que está compuesto por un número determinado de ciclos. La tarea del rol Control del entrenamiento en entorno compartido será garantizar que todos los agentes se encuentren siempre en el mismo episodio y en el mismo ciclo de su ejecución. Además tendrá que tener en cuenta que un agente puede no necesitar más ciclos o más episodios, por lo que el proceso ha de seguir dejándole atrás cuando sea necesario.

La forma de controlar la ejecución consistirá en esperar peticiones de episodio, dando paso a todos los agentes al mismo tiempo cuando tenga tantas peticiones como agentes necesita. Entonces esperará a tener tantas peticiones de ciclo como agentes necesita. Cuando las tenga todas, irá dando paso uno por uno a todos los agentes en orden aleatorio, esperando a que el primero acabe para dar paso al segundo y así sucesivamente.

#### **4.3.3. Roles ligados a la arquitectura**

Además, por las características de la plataforma de agentes elegida, JADE, y por las decisiones tomadas al respecto de la arquitectura propia del sistema, son necesarios los siguientes roles:

- **Registrador DF:** Registra a un agente en el agente DF de JADE. Un agente puede jugar este rol ninguna, una o varias veces, dependiendo de los servicios que ofrezca.

- **Buscador de agentes:** Busca agentes según su registro en el agente DF de JADE y solicita el uso de sus servicios. Un agente puede jugar este rol ninguna, una o varias veces, dependiendo de los agentes que necesite para su ejecución.
- **Servidor:** Acepta solicitudes de uso de otros agentes presentes en el sistema. Los tipos de agentes aceptados son configurables, así como la capacidad del agente como servidor.
- **Procesador:** Procesa información de algún tipo determinado. Este rol ejecuta una función de proceso por definir al pasar un tiempo configurable.
- **Finalizable:** Espera solicitudes para finalizar la ejecución de un rol determinado. Los roles que pueden finalizar la ejecución son configurables, así como las operaciones que llevan a la finalización del agente.

#### 4.4. Modelo de interacciones

El propósito del modelo de interacciones es completar el anterior modelo preliminar de interacciones, en el que se exponían las interacciones necesarias para el sistema sin comprometerse con una arquitectura en concreto.

El modelo de interacciones completo especifica todas las interacciones necesarias para la implementación del sistema multiagente. Se encuentra detallado en el Anexo II, a continuación se explican los cambios realizados frente al modelo preliminar de interacciones.

##### 4.4.1. Interacciones con la organización Estadística

Existen dos interacciones posibles con la organización estadística, y es la de enviar las estadísticas generadas durante el proceso de aprendizaje en entorno compartido o en entorno exclusivo. La organización se encarga internamente del proceso de dichas estadísticas, pero lo hará siempre que sea requerido por el Control del entrenamiento en entorno compartido.

##### 4.4.2. Interacciones con los roles arquitectónicos

Los nuevos roles aparecidos como consecuencia de la plataforma y la arquitectura elegidas también tienen sus propias interacciones.

- El rol **Servidor** permitirá recibir solicitudes de uso por parte de los agentes interesados en dichos servicios.
- El rol **RegistradorDF** registrará los servicios que ofrezca un agente en el agente DF de la plataforma JADE
- El rol **BuscadorAgentes** buscará agentes coincidan con una descripción dada en el agente DF de la plataforma JADE.
- El rol **Finalizable** aceptará peticiones de finalización por parte de otros agentes.



## 5. Diseño detallado

En este capítulo se describe detalladamente el sistema a construir, definiendo los procesos de aprendizaje necesarios, el modelo de agentes, el modelo de servicios y todos los protocolos presentes en las interacciones del sistema. Este capítulo se corresponde al Diseño Detallado propuesto por la metodología Gaia, añadiendo los apartados necesarios para detallar el proceso de aprendizaje.

### 5.1. Definición de los procesos de aprendizaje

En primer lugar se definen los dos procesos de aprendizaje que ejecutará el sistema. El primero de ellos dispondrá del entorno en exclusiva, no siendo molestado por ningún otro agente. En el segundo proceso, el entorno es compartido por todos los agentes.

#### 5.1.1. Proceso de entrenamiento en entorno exclusivo

El proceso de aprendizaje en entorno exclusivo usa el **algoritmo de Q-Learning** anteriormente explicado con las siguientes características:

- Política de **actualización determinista**  $Q(s, a) \leftarrow r_{s,a} + \gamma \max_{a'} Q(s', a')$  con el parámetro de descuento  $\gamma$  configurable por el usuario.
- Política de decisión de la acción a tomar  $\epsilon$  - **greedy** con el parámetro  $\epsilon$  configurable por el usuario.
- **Refuerzo positivo** de 1 al transitar a un estado meta y de 0 al transitar a un estado libre. Se podrán recibir refuerzos negativos de -1 al intentar transitar a un estado obstáculo según la configuración del usuario.
- El conjunto de movimientos es el de movimientos desde una casilla en las **cuatro direcciones posibles**. Adicionalmente, el usuario puede incluir el movimiento nulo entre los movimientos válidos.
- Episodios y ciclos del proceso configurables por el usuario.

Existirán tantos procesos de entrenamiento en entorno exclusivo como equipos diferentes haya en el sistema. Al final de cada uno de dichos procesos los agentes transferirán el conocimiento adquirido al entorno Repositorio.

#### 5.1.2. Proceso de entrenamiento en entorno compartido

El proceso de aprendizaje en entorno compartido usa el **algoritmo de Q-Learning** anteriormente explicado con las siguientes características:

- Política de actualización **no determinista**  

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha [r_{s,a} + \gamma \max_{a'} Q(s', a')]$$
 con parámetro de descuento  $\gamma$  configurable por el usuario y parámetro de enfriamiento  $\alpha$  configurable por el usuario.
- **Refuerzo positivo** de 1 al transitar a un estado meta y de 0 al transitar a un estado libre. Se podrán recibir refuerzos negativos de -1 al intentar transitar a un estado obstáculo según la configuración del usuario.
- El conjunto de movimientos es el de movimientos desde una casilla en las **cuatro direcciones** posibles. Adicionalmente, el usuario puede incluir el movimiento nulo entre los movimientos válidos.
- Episodios y ciclos del proceso configurables por el usuario.

### 5.1.3. *Conocimiento previo del entorno*

De esta forma, existen dos procesos de aprendizaje según el uso del entorno que se dé. El primero en entorno exclusivo y el segundo en entorno compartido. La primera técnica importante utilizada es **usar el conocimiento adquirido** en el primer proceso como semilla del segundo aprendizaje.

### 5.1.4. *Política de decisión de la acción en el proceso de entrenamiento en entorno compartido*

La política de decisión de la acción a tomar será la variación de la política  $\epsilon$  – *greedy* con probabilidades que serán modificadas según la evolución del entrenamiento explicada anteriormente (Análisis del sistema, política de decisión de acción  $\epsilon$  – *greedy* variable). Las probabilidades iniciales serán configuradas por el usuario. Concretamente, existirán tres tipos de estrategias de exploración:

- **Aleatoria.**
- Movimientos elegidos según el **conocimiento en entorno exclusivo** adquirido por los Jugadores en Entorno Exclusivo. Dichos movimientos llevarán a la meta por el camino más rápido sin tener en cuenta al resto de agentes del sistema.
- Movimientos elegidos según el **conocimiento en entorno compartido** adquirido por el propio Jugador en Entorno Compartido. Es decir, según la política que el agente está intentado aprender.

La probabilidad de elegir un movimiento de un tipo determinado será modificada según un incremento, entre unos máximos y mínimos. Tanto el incremento como los máximos y mínimos serán configurables por el usuario. De esta forma, el algoritmo para elegir que acción ejecutar será el siguiente

- Inicializar las probabilidades de elegir movimiento aleatorio ( $P_a$ ), elegir movimiento basado en el conocimiento en entorno exclusivo ( $P_e$ ) y elegir movimiento basado en el conocimiento adquirido en entorno compartido ( $P_c$ ) según los parámetros recibidos.
- Inicializar los incrementos de las probabilidades de elegir movimiento aleatorio ( $I_a$ ), elegir movimiento basado en el conocimiento en entorno exclusivo ( $I_e$ ) y elegir movimiento basado en el conocimiento adquirido en entorno compartido ( $I_c$ ) según los parámetros recibidos.
- Repetir (episodio de aprendizaje)
  - o Repetir (ciclo de aprendizaje)
    - Tomar un número aleatorio  $n$ .
    - Si  $n < P_a$ , elegir un movimiento aleatorio.
    - Si  $n > P_a$  y  $n < P_a + P_e$ , elegir un movimiento basado en el conocimiento en entorno exclusivo.
    - Si  $n > P_a + P_e$  y  $n < P_a + P_e + P_c$ , elegir un movimiento basado en el conocimiento en entorno compartido.
    - Ejecutar el movimiento.
  - o Hasta alcanzar el estado meta o el límite de ciclos por episodio.
  - o Si se ha alcanzado un estado meta, actualizar las probabilidades según
    - $P_a = P_a - I_a$
    - $P_e = P_e + I_e$
    - $P_c = P_c + I_c$
  - o Si no, actualizar las probabilidades según
    - $P_a = P_a + I_a$
    - $P_e = P_e - I_e$
    - $P_c = P_c - I_c$
  - o Ajustar las probabilidades.
- Hasta alcanzar el límite de episodios del experimento

**Ilustración 37 : Algoritmo de decisión de la estrategia de movimiento a seguir**

Por ejemplo, en la ilustración 29 podemos ver la evolución en un supuesto proceso de aprendizaje de 100 episodios con la siguiente configuración

- Probabilidad inicial de elegir un movimiento aleatorio: 0.4. Incremento de -0.004.
- Probabilidad inicial de elegir un movimiento según políticas en entorno exclusivo: 0.5. Incremento de -0.005.
- Probabilidad inicial de elegir un movimiento según política en entorno compartido: 0.0. Incremento de +0.109.

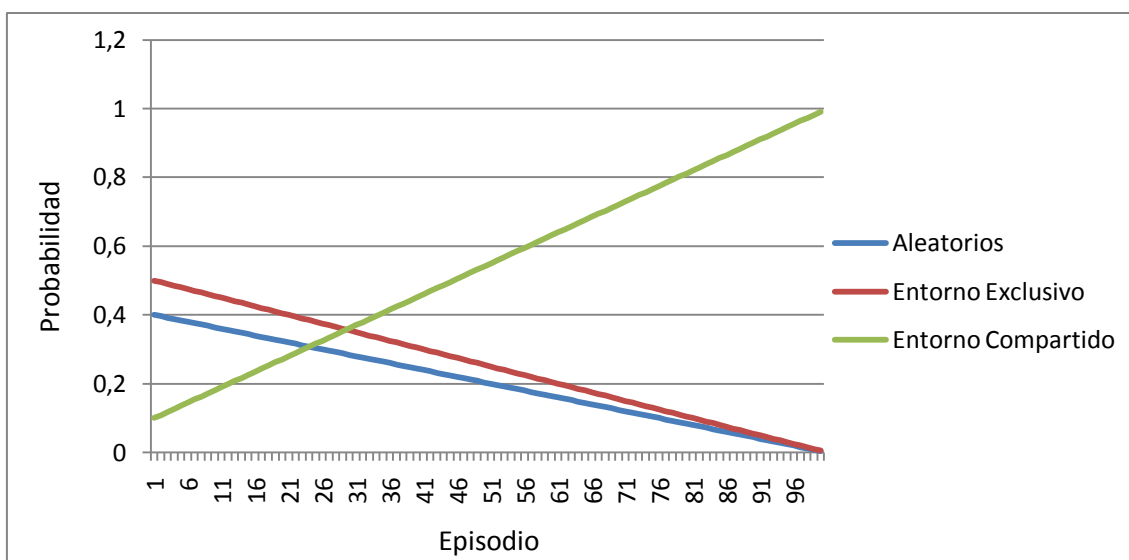


Ilustración 38 : Ejemplo de  $\epsilon$  - greedy variable

#### 5.1.5. Política de gestión del conocimiento en el proceso de entrenamiento en entorno compartido

Cada agente **enviará y pedirá conocimiento** contenido en el entorno Repositorio. Concretamente, al acabar cada episodio de entrenamiento en entorno compartido, enviará todas las tuplas adquiridas al agente encargado de la gestión del conocimiento.

En cuanto a la adquisición del conocimiento contenido en el entorno Repositorio, el agente pedirá la política contenida si se da alguna de las siguientes circunstancias:

- Si el número de ciclos usados en un episodio de aprendizaje supera un límite concreto. El límite será configurable por el usuario.
- Si ha pasado un número de episodios de aprendizaje sin pedir la política compartida. Dicho número será configurable por el usuario.

#### 5.1.6. Retardo del proceso de entrenamiento en entorno compartido

Adicionalmente, será posible **retardar el inicio del proceso** de los agentes secuencialmente. De esta forma, al inicio del proceso sólo habrá un agente sobre el entorno y cada determinado número de episodios, un nuevo agente entrará en el sistema. Dicho número de episodios será configurable por el usuario. El algoritmo será el siguiente

- Inicializar el retardo R según los parámetros.
- Inicializar el número de episodio N = 0.
- Repetir (proceso de aprendizaje)
  - Si  $N \bmod R = 0$  y quedan agentes por inicializar, inicializar el proceso de aprendizaje de un agente.

- |  |
|--|
| <ul style="list-style-type: none"><li>○ Iniciar un episodio de aprendizaje para todos los agentes inicializados.</li><li>○ N++</li><li>- Hasta que se alcance el límite de episodios</li></ul> |
|--|

Ilustración 39 : Algoritmo de retardo del proceso de aprendizaje

## 5.2. Definición del modelo de agentes

Según la metodología GAIA en el modelo de agentes se describen los agentes necesarios en el sistema, además de los roles que ejecutarán en el sistema y sus comportamientos asociados.

### 5.2.1. Organización Tablero

La organización encargada de gestionar el entorno Tablero estará formada por tres agentes, uno por cada tarea propia de la organización:

- **Agente estado:** Cumplirá con las tareas del rol Gestor de Estado, tramitando todas las posibles peticiones de estado.
- **Agente reinicio:** Cumplirá con las tareas del rol Gestor de Reinicio, tramitando todas las posibles peticiones de reinicio del estado en el entorno Tablero.
- **Agente movimiento:** Cumplirá con las tareas del rol Gestor de Movimiento, tramitando todas las posibles peticiones de acciones sobre el entorno Tablero.

Además, todos los agentes deberán:

- Registrarse en el agente DF de JADE, para que puedan ser encontrados por aquellos agentes que necesiten sus servicios.
- Aceptar peticiones de uso a través del rol servidor, para que los agentes que necesiten sus servicios puedan pedir permiso para utilizarlos.
- Aceptar peticiones de finalización de ejecución por parte de los agentes de control cuando la ejecución del sistema haya acabado.

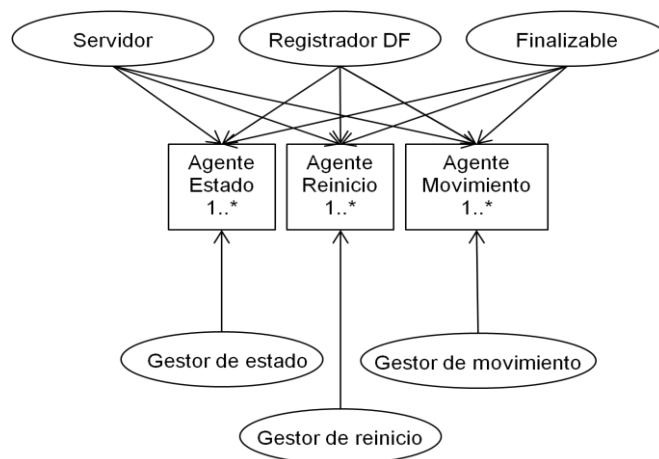


Ilustración 40 : Agentes de la organización Tablero y sus roles

### 5.2.2. Organización Estadística

La organización encargada de gestionar el entorno Estadístico estará formada por dos agentes, uno para cada una de las tareas propias de la organización:

- **Agente Gestor Estadístico:** Recoge las estadísticas de los agentes jugadores y las insertan en el entorno Estadístico.
- **Agente Procesador Estadístico:** Procesa la actualización de la gráfica. Dicho proceso será ejecutado cada periodo de tiempo dado por parámetro y por posibles peticiones expresas.

Además todos los agentes deberán:

- Registrarse en el agente DF de JADE, para que puedan ser encontrados por aquellos agentes que necesiten sus servicios.
- Aceptar peticiones de uso a través del rol servidor, para que los agentes que necesiten sus servicios puedan pedir permiso para utilizarlos.
- Aceptar peticiones de finalización de ejecución por parte de los agentes de control cuando la ejecución del sistema haya acabado.

Adicionalmente, el Agente Procesador Estadístico cumplirá el rol Procesador, que hará al agente ejecutar sus funciones cada cierto periodo de tiempo dado por parámetro.

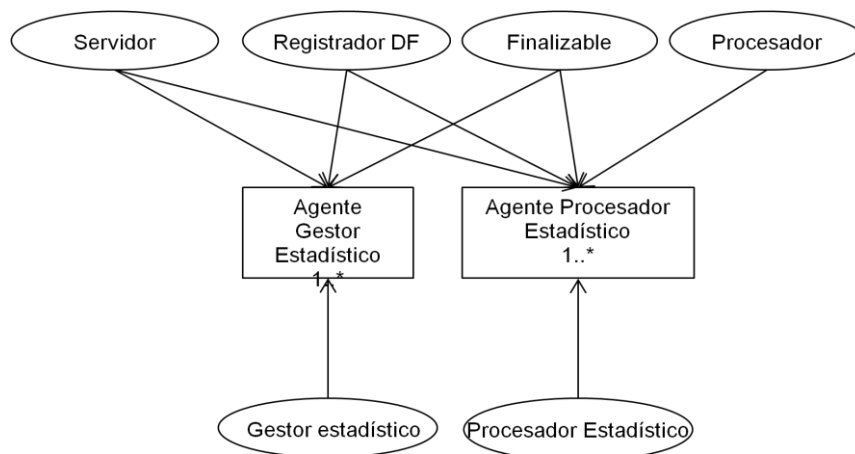


Ilustración 41 : Agentes de la organización Estadística y sus roles

### 5.2.3. Organización Repositorio

La organización encargada de gestionar el entorno Repositorio estará formada por tres agentes, uno para cada tarea propia de la organización.

- **Agente Gestor de Conocimiento Compartido:** Recibe las políticas de movimientos generadas en entornos compartidos y gestiona las peticiones de dichas políticas por parte de los agentes jugadores.
- **Agente Gestor de Conocimiento Exclusivo:** Recibe las políticas de movimientos generadas en entornos exclusivos y gestiona las peticiones de los movimientos incluidos en dichas políticas por parte de los agentes jugadores.
- **Agente Procesador de Conocimiento:** Procesa la actualización del conocimiento contenido en el entorno Repositorio. Dicho proceso será ejecutado cada periodo de tiempo dado por parámetro.

Además, todos los agentes deberán:

- Registrarse en el agente DF de JADE, para que puedan ser encontrados por aquellos agentes que necesiten sus servicios.
- Aceptar peticiones de finalización de ejecución por parte de los agentes de control cuando la ejecución del sistema haya acabado.

Por otro lado, Agente Gestor Conocimiento Compartido y Agente Gestor Conocimiento Exclusivo implementarán el rol Servidor para que los usuarios que necesiten sus servicios puedan solicitar su uso.

Por último, el Agente Procesador Conocimiento implementará el rol Procesador, que hará al agente ejecutar sus funciones cada cierto periodo de tiempo dado por parámetro.

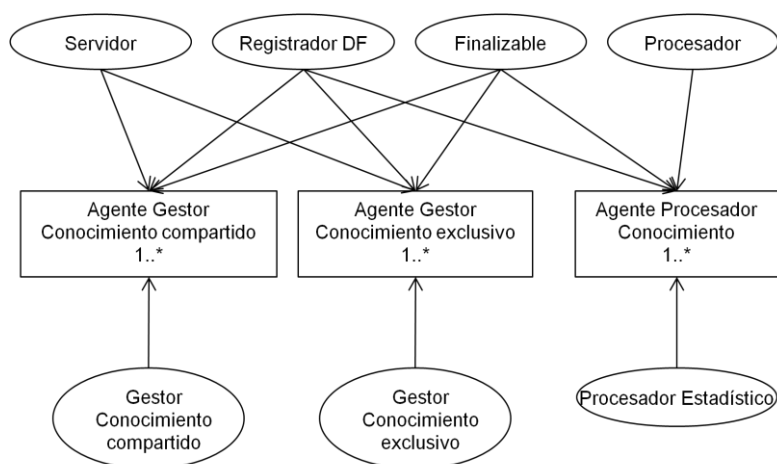


Ilustración 42 : Agentes de la organización Repositorio y sus roles

#### 5.2.4. Jugadores en Entorno Exclusivo

La organización de jugadores que ejecutan sus acciones en un entorno exclusivo tiene un único agente, Agente Jugador Entorno Exclusivo, del que existirán tantas instancias como jugadores de este tipo sean necesarios. El **Agente Jugador en Entorno Exclusivo** debe:

- Registrarse en el agente DF de JADE, para que pueda ser encontrado por otros agentes.
- Aceptar peticiones de finalización de ejecución por parte de los agentes de control cuando la ejecución del sistema haya acabado.
- Buscar los agentes que necesite para llevar a cabo todas sus funciones.

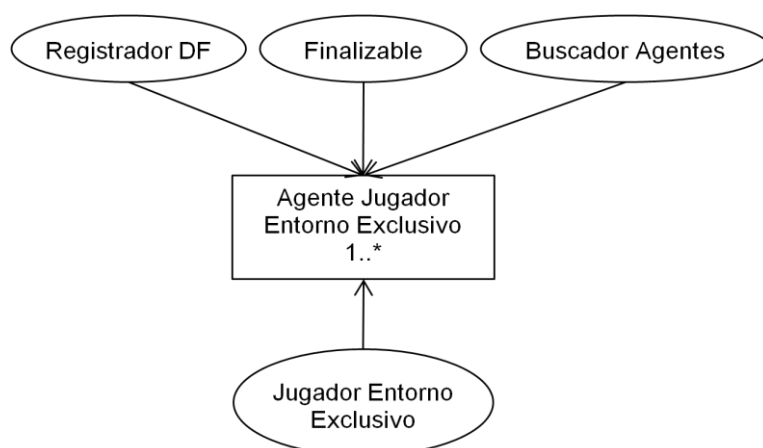


Ilustración 43 : Agente Jugador en Entorno Exclusivo y sus roles



### 5.2.5. Jugadores en Entorno Compartido

La organización de jugadores que ejecutan sus acciones en un entorno compartido con otros agentes tiene un único agente, Agente Jugador Entorno Compartido del que existirán tantas instancias como jugadores de este tipo sean necesarios. El **Agente Jugador en Entorno Compartido** debe:

- Registrarse en el agente DF de JADE, para que pueda ser encontrado por otros agentes.
- Aceptar peticiones de finalización de ejecución por parte de los agentes de control cuando la ejecución del sistema haya acabado.
- Buscar los agentes que necesite para llevar a cabo todas sus funciones.

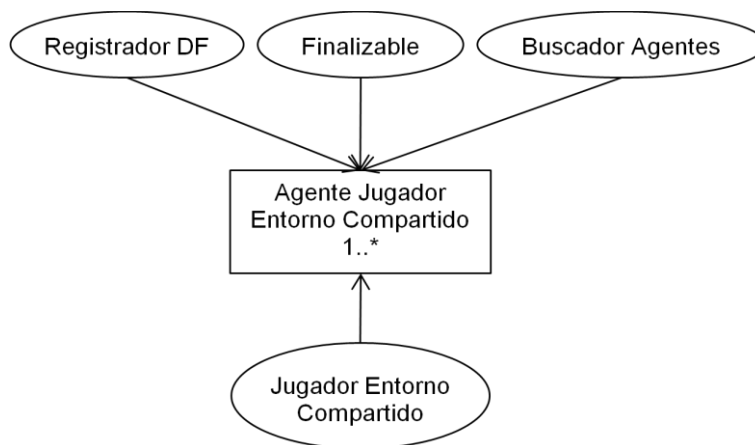


Ilustración 44 : Agente Jugador en Entorno Compartido y sus roles

### 5.2.6. Organización de Control

La organización Control está compuesta por dos agentes únicos en el sistema, un primero encargado de controlar los procesos de entrenamiento en entorno exclusivo y un segundo encargado de controlar los procesos de entrenamiento en entorno compartido. Además, ambos deben:

- Registrarse en el agente DF de JADE, para que puedan ser encontrados por aquellos agentes que necesiten sus servicios.
- Aceptar peticiones de uso a través del rol servidor, para que los agentes que necesiten sus servicios puedan pedir permiso para utilizarlos.

Adicionalmente, el Agente Control del Entrenamiento en Entorno Exclusivo debe:

- Buscar los agentes que necesita para su ejecución. Concretamente, debe buscar al otro agente de control, Agente Control del Entrenamiento en Entorno Compartido, para poder informarle de que el entrenamiento en entorno exclusivo ha acabado cuando sea preciso.
- Aceptar peticiones de finalización del otro agente de control, Agente Control del Entrenamiento en Entorno Compartido,

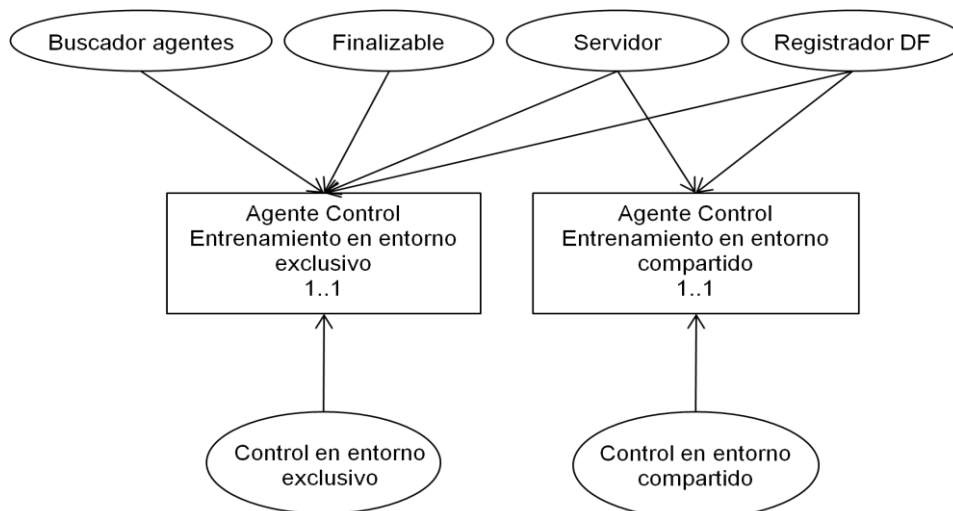


Ilustración 45 : Agentes de la organización Control y sus roles

### 5.3. Definición del modelo de servicios

El modelo de servicios de GAIA sirve para definir todos los servicios ofrecidos por el sistema, así como los roles que los ofrecen. En el anexo V se puede encontrar el modelo completo.

Servicio	Rol(es) iniciador(es)	Rol servidor
Obtener estado	Jugador en Entorno Exclusivo y Jugador en Entorno Compartido	Gestor Estado
Ejecutar movimiento	Jugador en Entorno Exclusivo y Jugador en Entorno Compartido	Gestor Movimiento
Pedir reinicio	Jugador en Entorno Exclusivo y Jugador en Entorno Compartido	Gestor Reinicio
Enviar estadísticas entorno exclusivo	Jugador en Entorno Exclusivo.	Gestor Estadístico
Enviar estadísticas entorno compartido	Jugador en Entorno Compartido	Gestor Estadístico
Actualizar gráfica	Control del entrenamiento en entorno compartido	Procesador estadístico
Enviar política exclusiva	Jugador en Entorno Exclusivo	Gestor de Conocimiento en Entorno Exclusivo

Enviar política compartida	Jugador en Entorno Compartido	Gestor de Conocimiento en Entorno Compartido
Obtener mejor movimiento en entorno exclusivo	Jugador en Entorno Compartido	Gestor de Conocimiento en Entorno Exclusivo
Obtener política compartida	Jugador en Entorno Compartido	Gestor de Conocimiento en Entorno Compartido
Solicitud de uso	Jugador en Entorno Exclusivo y Jugador en Entorno Compartido	Servidor
Pedir registro	RegistradorDF	Agente DF
Finalizar ejecución	Control del entrenamiento en entorno exclusivo o control del entrenamiento en entorno compartido	Finalizable

Tabla 5: Modelo de servicios

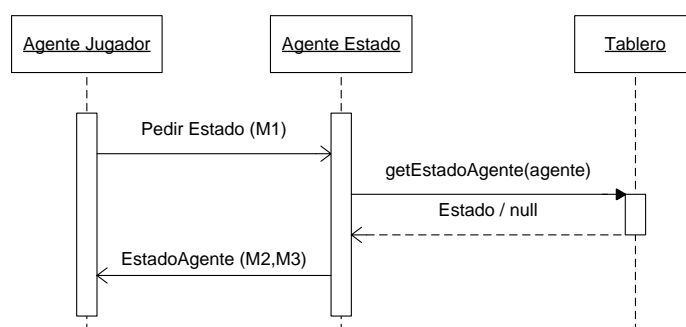
## 5.4. Definición de los protocolos

En el presente apartado se definen los protocolos que se usarán para todas las interacciones y servicios existentes en el sistema.

Existen dos formas de comunicación entre las distintas entidades. Por un lado la **comunicación asíncrona** que se da entre los agentes del sistema. Esta comunicación consiste en mensajes ACL. Se indica entre paréntesis los distintos mensajes que pueden ser enviados en cada comunicación. Por otro lado la **comunicación síncrona** que se da entre agentes y los diferentes entornos. Esta comunicación consiste en llamadas a funciones y se indica entre paréntesis los parámetros que reciben dichas funciones.

### 5.4.1. Pedir Estado

El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente jugador (que puede ser Jugador en Entorno Exclusivo o Jugador en Entorno Compartido) pide conocer su estado dentro del entorno Tablero



Protocolo 1 : Pedir Estado

Los motivos por los que el Agente Estado puede responder con un mensaje **EstadoDesconocido** (M4, M5) son:

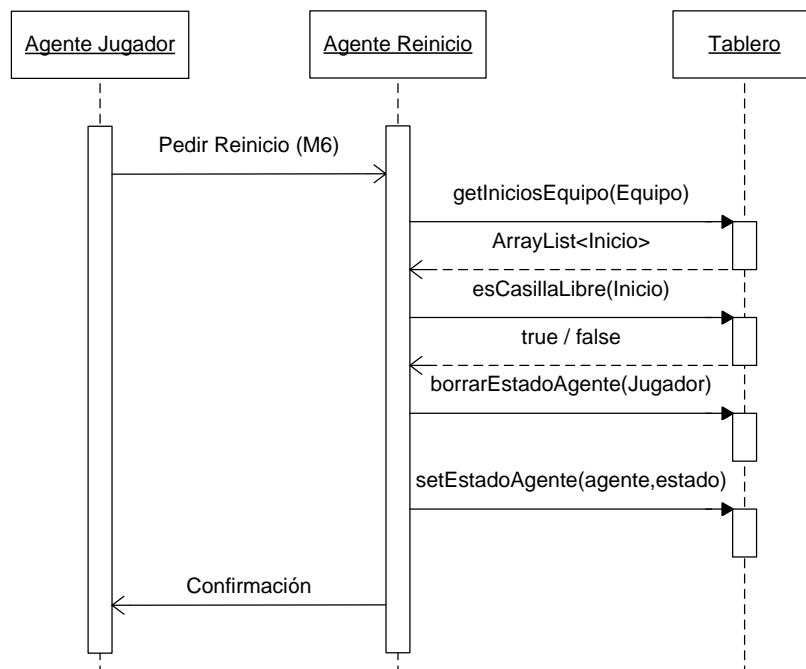
- 1) Que la posición del agente sea desconocida, esto es, que no se encuentre en el Entorno.

Los motivos por los que el agente Entorno puede responder **AgenteNoAutorizado** (M33) son:

- 1) El agente que ha enviado el mensaje no es el mismo que para el que se solicita conocer el estado. Un agente sólo puede solicitar conocer su propio estado.
- 2) El agente que pide conocer su estado no se encuentra entre los agentes autorizados para el uso de este agente Estado.
- 3) El agente que ha solicitado el conocer su estado no es de un tipo permitido.

#### 5.4.2. Pedir Reinicio

El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente jugador (que puede ser Jugador en Entorno Exclusivo o Jugador en Entorno Compartido) pide que su estado sea reiniciado en el entorno Tablero.



Protocolo 2 : Pedir Reinicio

Los motivos por los que el Agente Reinicio puede responder con un mensaje **ReinicioIncorrecto** (M7) son:

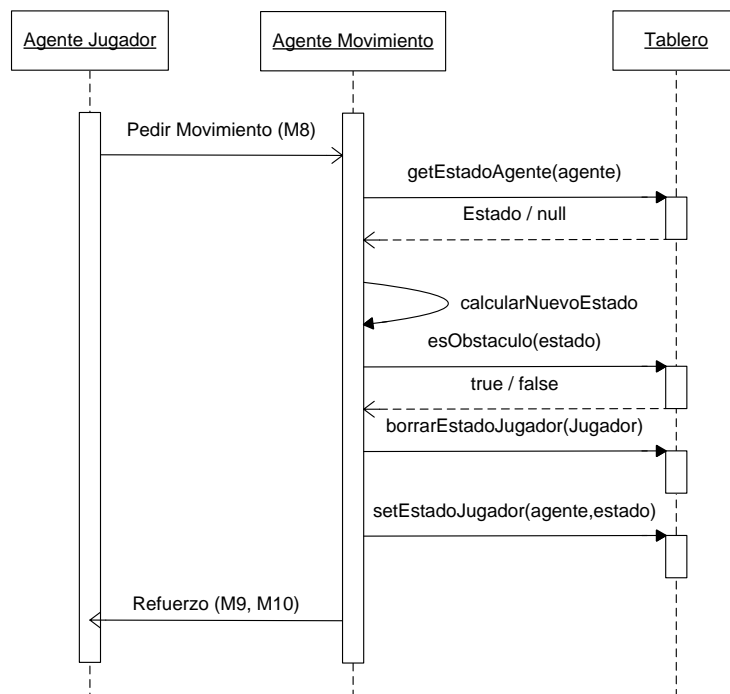
- 1) No se han encontrado casillas de inicio libres. El agente debe esperar y volver a pedir el reinicio.

Los motivos por los que el agente Entorno puede responder **AgenteNoAutorizado** (M33) son:

- 1) El agente que ha enviado el mensaje no es el mismo que para el que se solicita el reinicio. Un agente sólo puede solicitar su propio reinicio.
- 2) El agente que pide reiniciar su estado no se encuentra entre los agentes autorizados para el uso de este agente Reinicio.
- 3) El agente que ha solicitado el reinicio no es de un tipo permitido.

#### 5.4.3. Pedir Movimiento

El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente jugador (que puede ser Jugador en Entorno Exclusivo o Jugador en Entorno Compartido) pide ejecutar una acción dentro del entorno Tablero



Protocolo 3 : Pedir Movimiento

Los motivos por los que el Agente Movimiento puede responder con un mensaje **MovimientoIncorrecto** (M10) son:

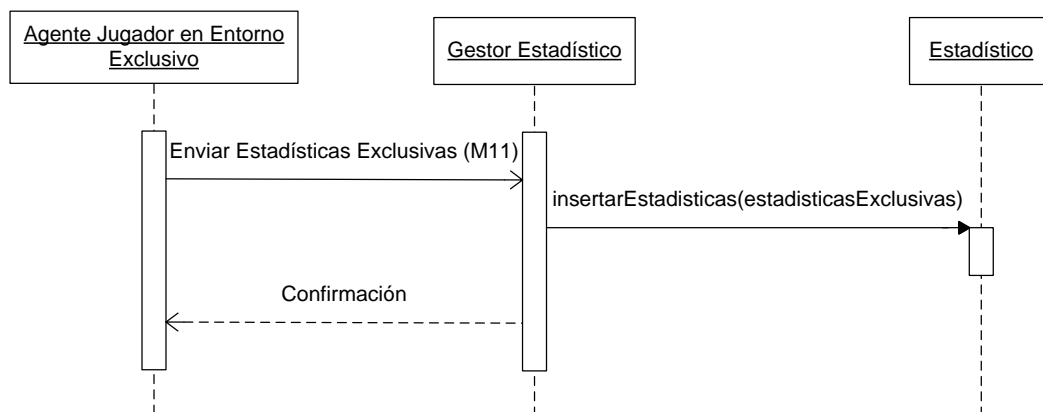
- 1) Que la posición del agente sea desconocida.
- 2) Que el movimiento solicitado no esté entre los movimientos autorizados.
- 3) Que el movimiento haya dado como resultado un choque contra un obstáculo del Tablero o bien contra otro Jugador. Recibirá entonces un refuerzo negativo.

Los motivos por los que el agente Movimiento puede responder **AgenteNoAutorizado** (M33) son:

- 1) El agente que ha enviado el mensaje no es el mismo que para el que se solicita el movimiento. Un agente sólo puede solicitar su propio reinicio.
- 2) El agente que pide reiniciar su estado no se encuentra entre los agentes autorizados para el uso de este agente Movimiento.
- 3) El agente que ha solicitado el movimiento no es de un tipo permitido.

#### 5.4.4. *Enviar Estadísticas Entorno Exclusivo*

El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Jugador en Entorno Exclusivo envía las estadísticas generadas durante un episodio de aprendizaje.



**Protocolo 4 : Enviar Estadísticas Entorno Exclusivo**

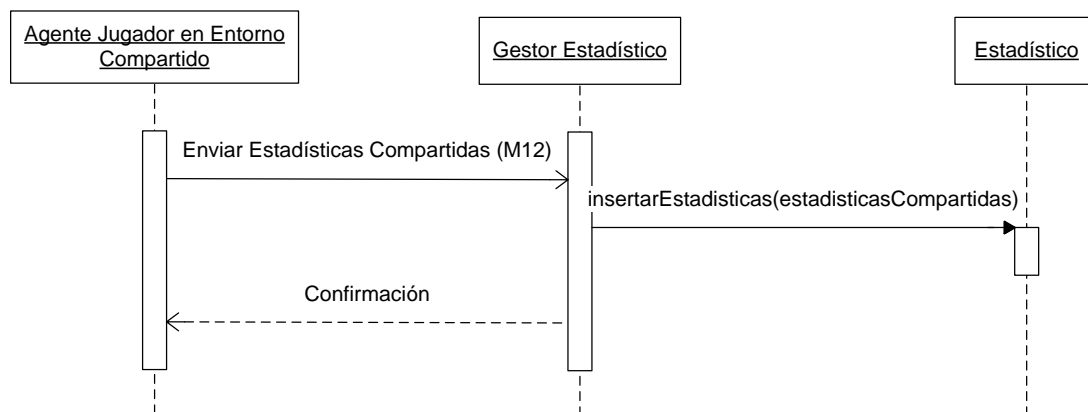
Los motivos por los que el Gestor Estadístico puede responder **AgenteNoAutorizado** (M33) son:

- 1) El agente que ha enviado el mensaje no es el mismo que el que ha generado las estadísticas. Un agente sólo puede enviar estadísticas de si mismo.

- 2) El agente que manda las estadísticas no se encuentra entre los agentes autorizados para el uso de este Gestor Estadístico.
- 3) El agente que ha enviado las estadísticas no es de un tipo permitido. Sólo pueden enviar estadísticas de episodios de entrenamiento en entorno compartido agentes Jugador en Entorno Exclusivo.

#### 5.4.5. *Enviar Estadísticas Entorno Compartido*

El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Jugador en Entorno Compartido envía las estadísticas generadas durante un episodio de aprendizaje.



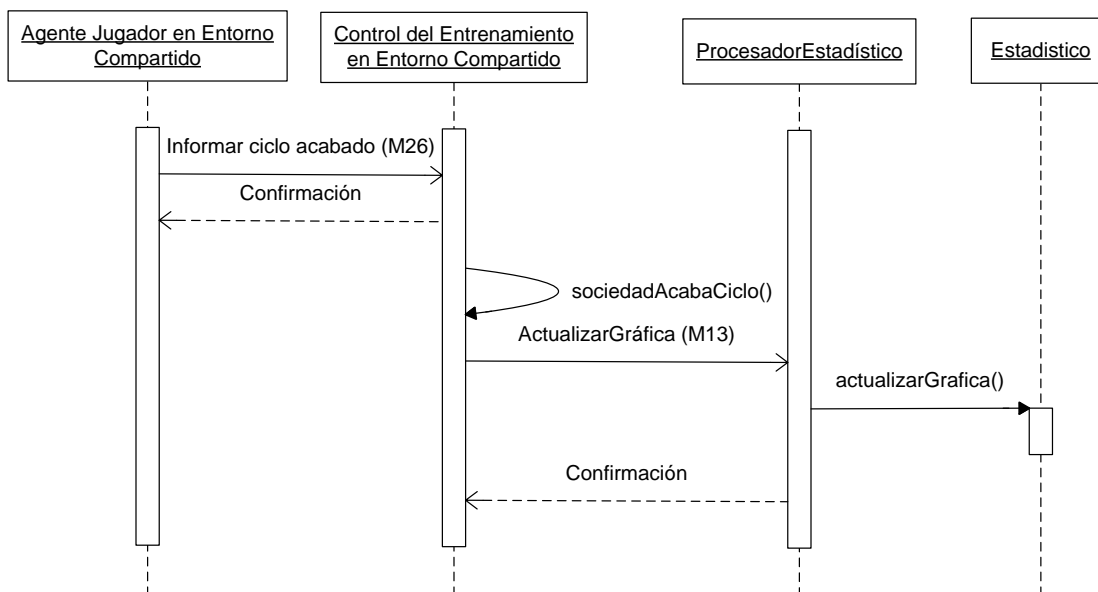
**Protocolo 5 : Enviar Estadísticas Entorno Compartido**

Los motivos por los que el Gestor Estadístico puede responder **AgenteNoAutorizado** (M33) son:

- 1) El agente que ha enviado el mensaje no es el mismo que el que ha generado las estadísticas. Un agente sólo puede enviar estadísticas de si mismo.
- 2) El agente que manda las estadísticas no se encuentra entre los agentes autorizados para el uso de este Gestor Estadístico.
- 3) El agente que ha enviado las estadísticas no es de un tipo permitido. Sólo pueden enviar estadísticas de episodios de entrenamiento en entorno compartido agentes Jugador en Entorno Compartido.

#### 5.4.6. *Actualizar gráfica*

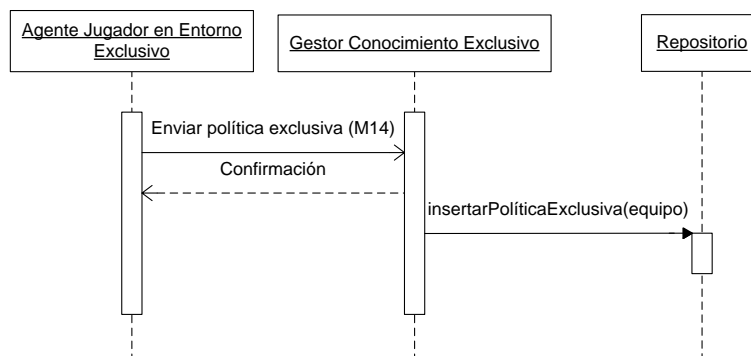
El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Control del Entrenamiento en Entornos Compartidos pide que sea actualizada la gráfica que muestra las estadísticas de los procesos de aprendizaje al usuario.



**Protocolo 6 : Actualizar gráfica**

#### 5.4.7. Enviar política exclusiva

El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Jugador en Entorno Exclusivo envía la política de acciones que ha aprendido durante su proceso de aprendizaje.



**Protocolo 7 : Enviar política exclusiva**

Los motivos por los que el Gestor Conocimiento Exclusivo puede responder **AgenteNoAutorizado** (M33) son:

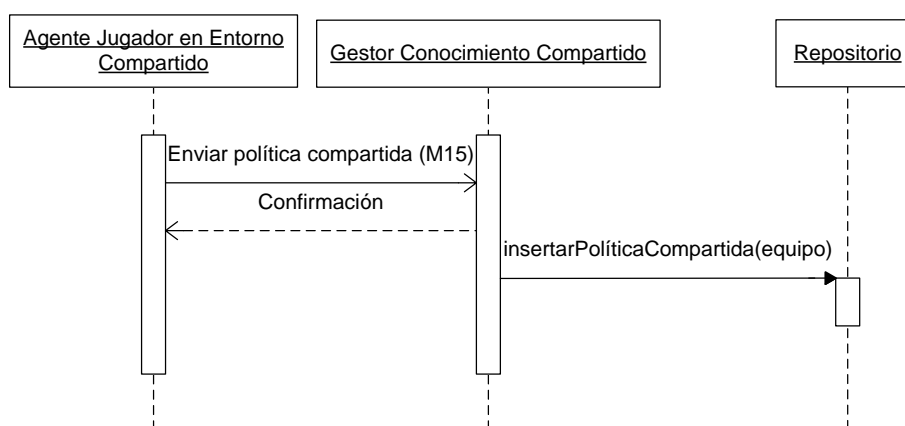
- 1) El agente que ha enviado el mensaje no es el mismo que el que ha generado el conocimiento. Un agente sólo puede enviar conocimiento adquirido por si mismo.



- 2) El agente que manda la política no se encuentra entre los agentes autorizados para el uso de este Gestor de Conocimiento Exclusivo.
- 3) El agente que ha enviado el conocimiento no es de un tipo permitido. Sólo pueden enviar conocimiento agentes Jugador en Entorno Exclusivo.

#### 5.4.8. *Enviar política compartida*

El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Jugador en Entorno Compartido envía la política de acciones que ha aprendido durante alguna parte de su proceso de aprendizaje.



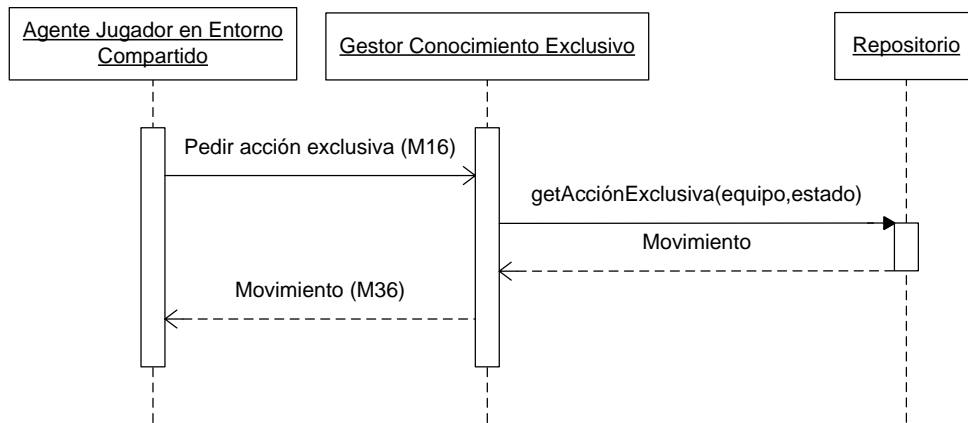
**Protocolo 8 : Enviar política compartida**

Los motivos por los que el Gestor Conocimiento Compartido puede responder **AgenteNoAutorizado** (M33) son:

- 1) El agente que ha enviado el mensaje no es el mismo que el que ha generado el conocimiento. Un agente sólo puede enviar conocimiento adquirido por si mismo.
- 2) El agente que manda la política no se encuentra entre los agentes autorizados para el uso de este Gestor de Conocimiento Compartido.
- 3) El agente que ha enviado el conocimiento no es de un tipo permitido. Sólo pueden enviar conocimiento agentes Jugador en Entorno Compartido.

#### 5.4.9. *Pedir mejor movimiento en entorno exclusivo*

El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Jugador en Entorno Compartido pide conocer la mejor acción según la política de acciones en entorno exclusivo.



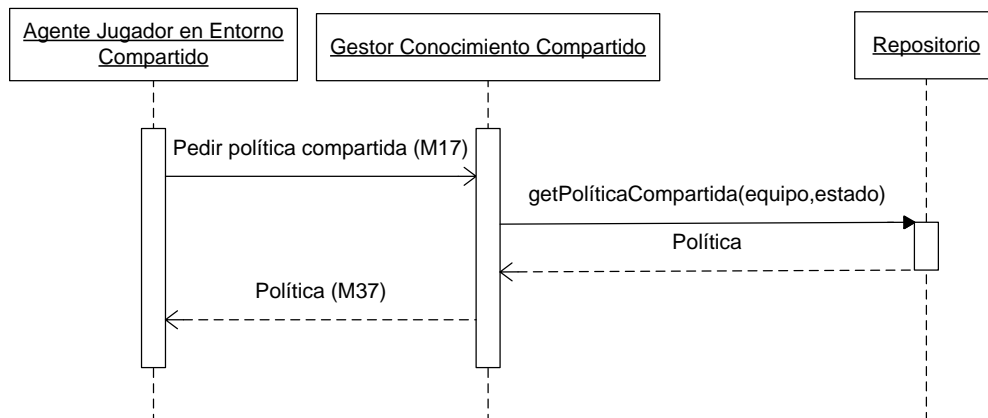
**Protocolo 9 : Pedir mejor movimiento en entorno exclusivo**

Los motivos por los que el Gestor Conocimiento Exclusivo puede responder **AgenteNoAutorizado** (M33) son:

- 1) El agente que solicita el conocimiento no se encuentra entre los agentes autorizados para el uso de este Gestor de Conocimiento Exclusivo.
- 2) El agente que solicita el conocimiento no es de un tipo permitido. Sólo pueden pedir conocimiento agentes Jugador en Entorno Compartido.

#### 5.4.10. Pedir política compartida

El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Jugador en Entorno Compartido pide conocer la política de acciones en entorno compartido para su equipo.



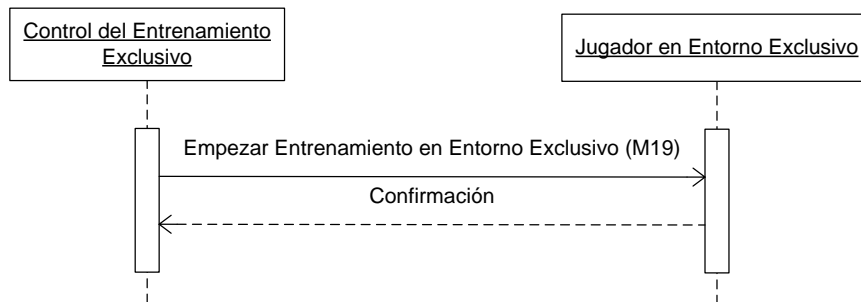
**Protocolo 10 : Pedir política compartida**

Los motivos por los que el Gestor Conocimiento Compartido puede responder **AgenteNoAutorizado** son:

- 1) El agente que solicita el conocimiento no se encuentra entre los agentes autorizados para el uso de este Gestor de Conocimiento Compartido.
- 2) El agente que solicita el conocimiento no es de un tipo permitido. Sólo pueden pedir conocimiento agentes Jugador en Entorno Compartido.

#### 5.4.11. Empezar entrenamiento exclusivo

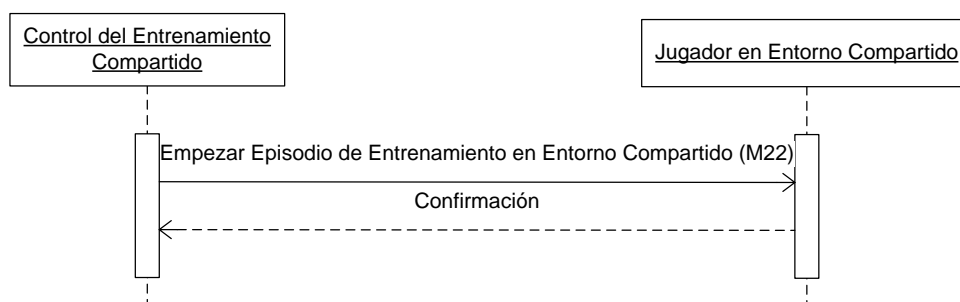
El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Control del Entrenamiento en Entorno Exclusivo autoriza a un agente Jugador en Entorno Exclusivo a empezar su proceso de aprendizaje.



Protocolo 11 : Empezar entrenamiento exclusivo

#### 5.4.12. Empezar episodio de entrenamiento compartido

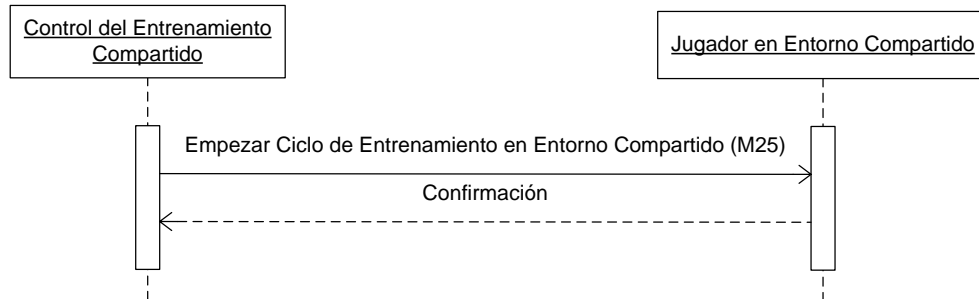
El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Control del Entrenamiento en Entorno Compartido autoriza a un agente Jugador en Entorno Compartido a empezar un episodio de su proceso de aprendizaje.



Protocolo 12 : Empezar episodio de entrenamiento compartido

#### 5.4.13. Empezar ciclo de entrenamiento compartido

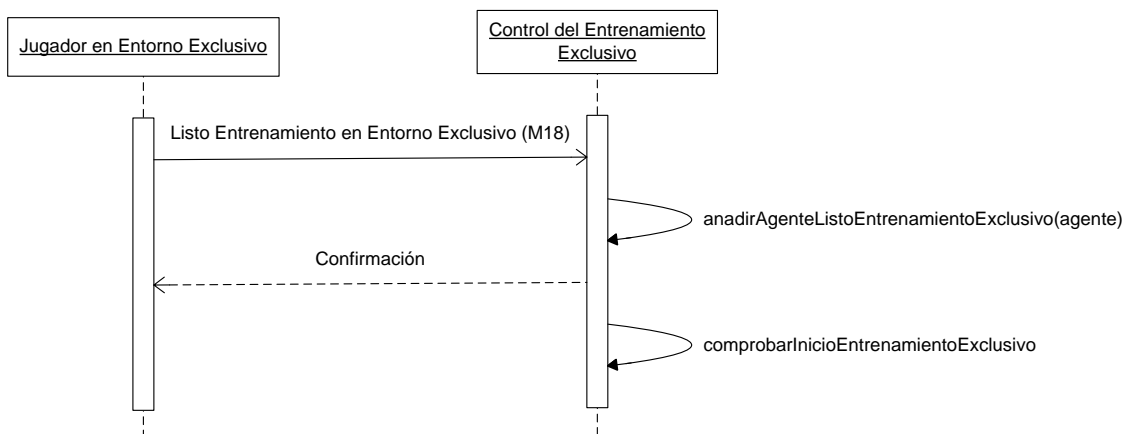
El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Control del Entrenamiento en Entorno Compartido autoriza a un agente Jugador en Entorno Compartido a empezar un ciclo de su proceso de aprendizaje.



Protocolo 13 : Empezar ciclo de entrenamiento compartido

#### 5.4.14. Listo para entrenamiento exclusivo

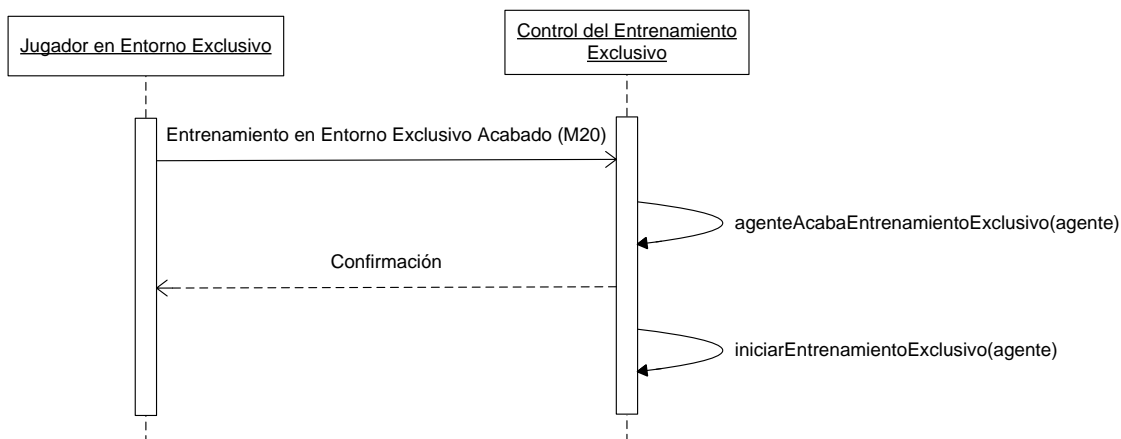
El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Jugador en Entorno Exclusivo informa al agente Control del Entrenamiento en Entorno Exclusivo de que se encuentra preparado para empezar su proceso de aprendizaje.



Protocolo 14 : Listo para entrenamiento exclusivo

#### 5.4.15. Entrenamiento exclusivo acabado

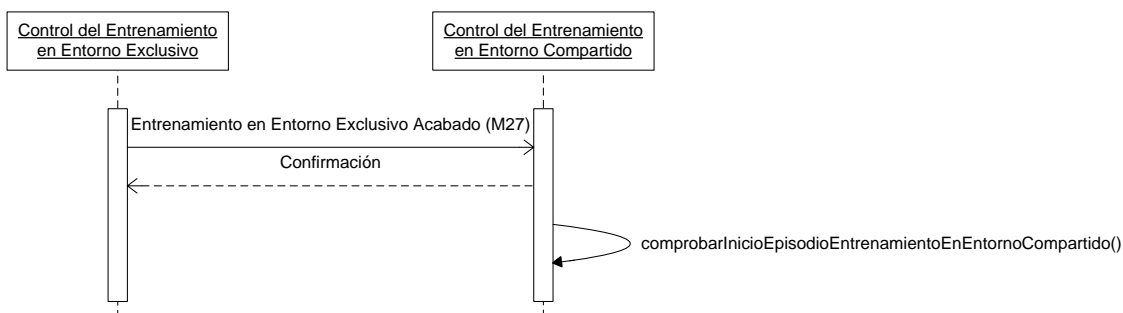
El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Jugador en Entorno Exclusivo informa al agente Control del Entrenamiento en Entorno Exclusivo de que ha acabado su proceso de aprendizaje.



Protocolo 15 : Entrenamiento exclusivo acabado

#### 5.4.16. Informar entrenamiento en entorno exclusivo acabado

El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando el agente Control del Entrenamiento en Entorno Exclusivo informa al agente Control del Entrenamiento en Entorno Compartido de que el proceso de aprendizaje en entorno exclusivo ha acabado.

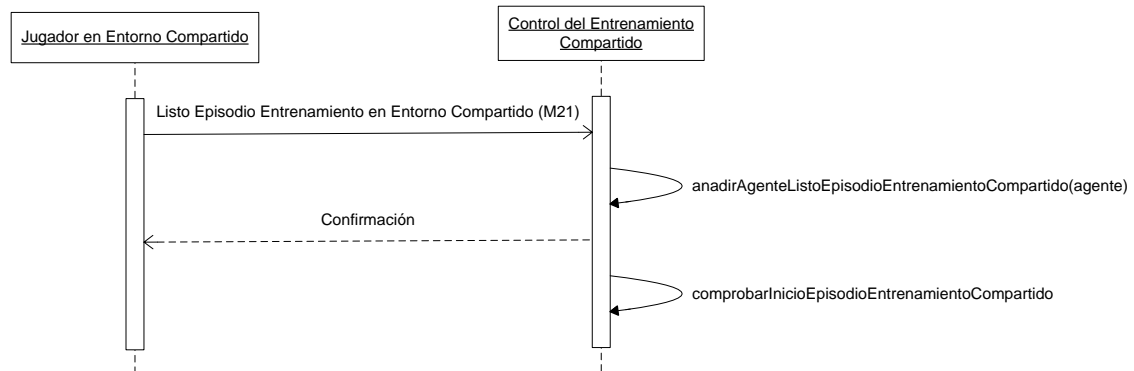


Protocolo 16 : Informar entrenamiento en entorno exclusivo acabado

#### 5.4.17. Listo para episodio de entrenamiento compartido

El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Jugador en Entorno Compartido informa al agente Control del Entrenamiento

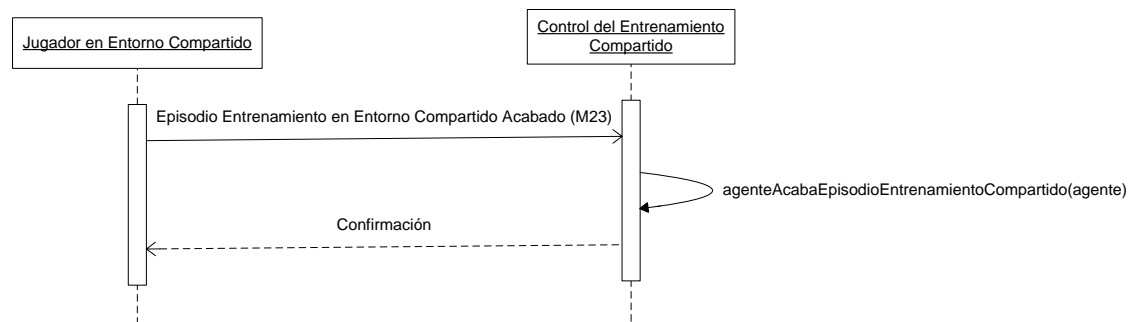
en Entorno Compartido de que se encuentra preparado para empezar un episodio de su proceso de aprendizaje.



**Protocolo 17 : Listo para episodio de entrenamiento compartido**

#### 5.4.18. Acabado episodio de entrenamiento compartido

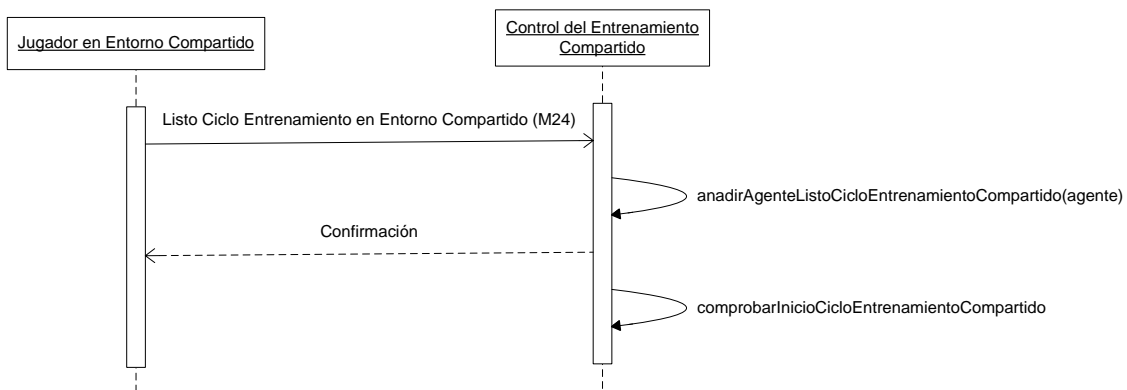
El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Jugador en Entorno Compartido informa al agente Control del Entrenamiento en Entorno Compartido de que ha acabado un episodio de su proceso de aprendizaje.



**Protocolo 18 : Acabado episodio de entrenamiento compartido**

#### 5.4.19. Listo para ciclo de entrenamiento compartido

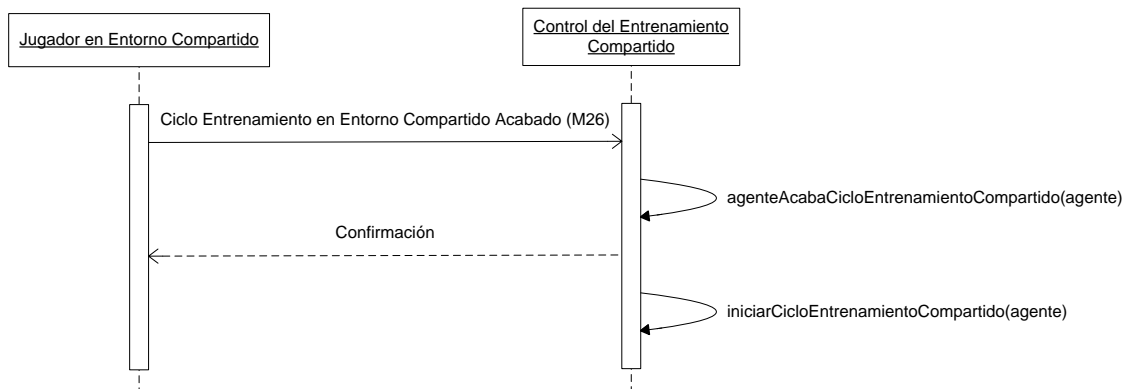
El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Jugador en Entorno Compartido informa al agente Control del Entrenamiento en Entorno Compartido de que se encuentra preparado para empezar un ciclo de su proceso de aprendizaje.



**Protocolo 19 : Listo para ciclo de entrenamiento compartido**

#### 5.4.20. Acabado ciclo de entrenamiento en entorno compartido

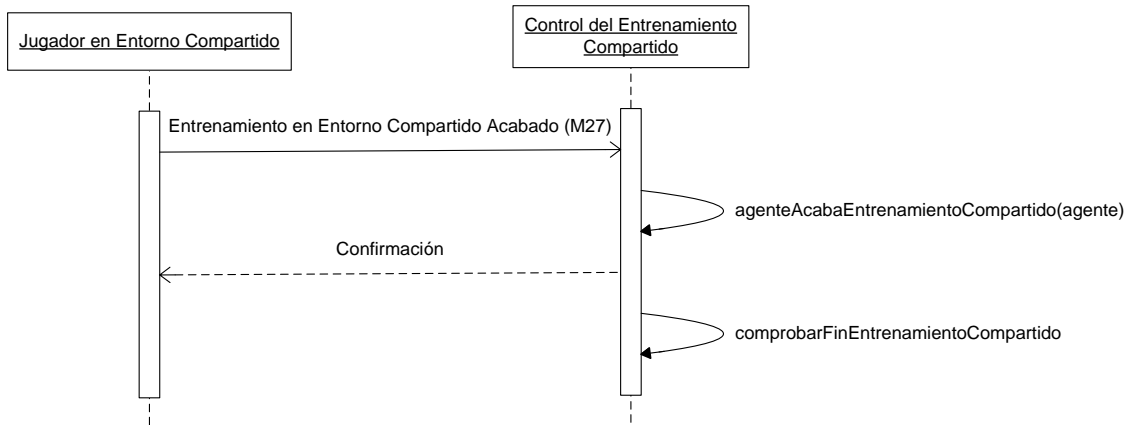
El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Jugador en Entorno Compartido informa al agente Control del Entrenamiento en Entorno Compartido de que ha acabado un ciclo de su proceso de aprendizaje.



**Protocolo 20 : Acabado ciclo de entrenamiento en entorno compartido**

#### 5.4.21. Entrenamiento en entorno compartido acabado

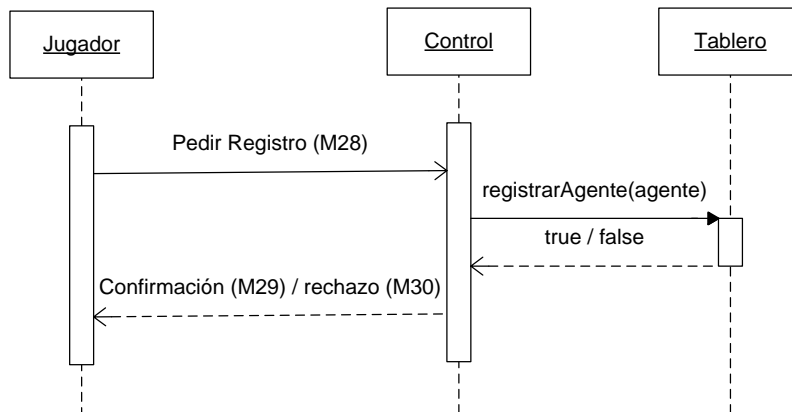
El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Jugador en Entorno Compartido informa al agente Control del Entrenamiento en Entorno Compartido de que ha acabado su proceso de aprendizaje.



**Protocolo 21 : Entrenamiento en entorno compartido acabado**

#### 5.4.22. Pedir registro en entorno Tablero

El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente jugador (que puede ser Jugador en Entorno Compartido o Jugador en Entorno Exclusivo) solicita al agente control (que puede ser Control del Entrenamiento en Entorno Exclusivo o Control del Entrenamiento en Entorno Compartido) ser registrado en el entorno Tablero.



**Protocolo 22 : Pedir registro en entorno Tablero**

Los motivos por los que el agente de control puede responder **AgenteNoRegistrado** (M29) son:

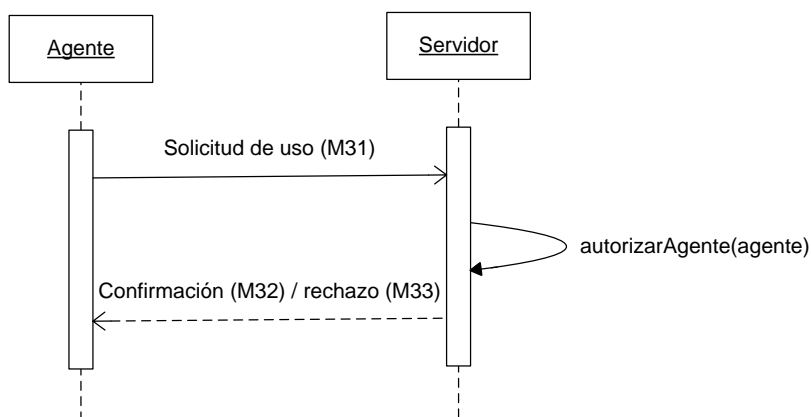
- 1) El agente que ha enviado el mensaje no es el mismo que para el que se solicita el registro. Un agente sólo puede solicitar registro para sí mismo.



- 2) El agente que ha solicitado el registro no es de un tipo permitido. Sólo pueden ser registrados agentes de tipo Jugador en Entorno Compartido o Jugador en Entorno Exclusivo.
- 3) El Tablero no admite más agentes del tipo que solicita el registro.

#### 5.4.23. Solicitud de Uso

El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente solicita autorización para usar los servicios de un agente Servidor.



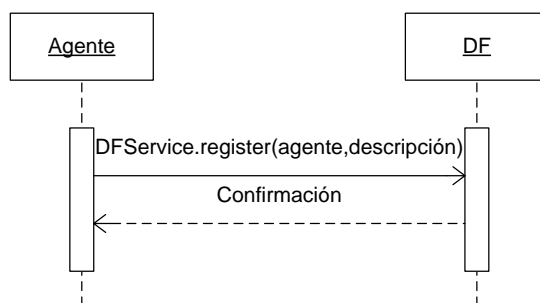
**Protocolo 23 : Solicitud de Uso**

Los motivos por los que el Servidor puede responder **AgenteNoAutorizado** (M33) son:

- 1) El agente que ha enviado el mensaje no es el mismo que para el que se solicita autorización. Un agente sólo puede solicitar autorización para sí mismo.
- 2) El agente que ha solicitado la autorización no es de un tipo permitido.
- 3) El agente Control no admite más agentes del tipo que solicita la autorización.

#### 5.4.24. Pedir registro

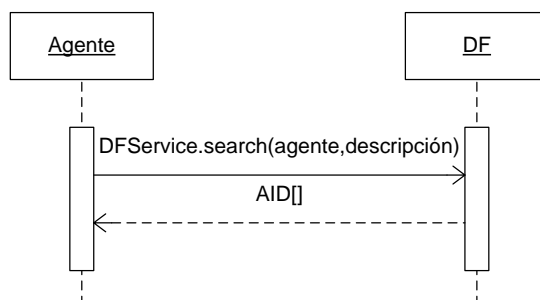
El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente se registra en el agente DF de JADE.



**Protocolo 24 : Pedir registro**

#### **5.4.25. Encontrar agente**

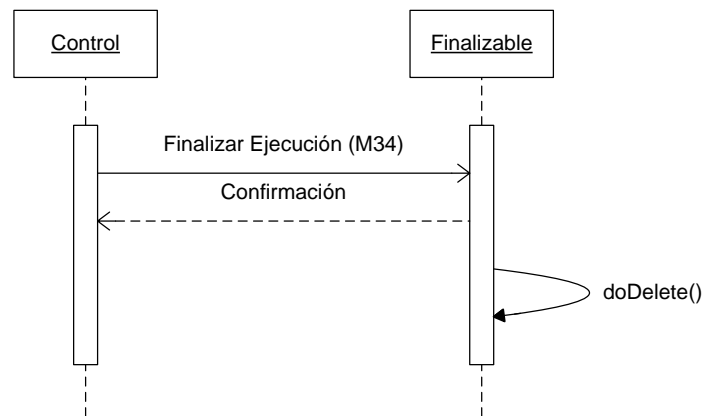
El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente realiza una búsqueda en el agente DF de JADE.



**Protocolo 25 : Encontrar agente**

#### **5.4.26. Finalizar Ejecución**

El siguiente diagrama contiene la secuencia de mensajes y llamadas que se produce cuando un agente Control (que puede ser Control del Entrenamiento en Entorno Exclusivo o Control del Entrenamiento en Entorno Compartido) solicita a otro agente Finalizable que finalice su ejecución.



#### Protocolo 26 : Finalizar Ejecución

Los motivos por los que el agente Finalizable puede responder **EjecuciónNoFinalizada** (M35) son:

- 1) El agente que ha enviado el mensaje no es un agente de Control.
- 2) El agente que ha enviado el mensaje no es el mismo que ordena el fin de la ejecución.
- 3) El agente que solicita que acabe la ejecución no es el mismo que el que recibe el mensaje.



## 6. Experimentos

En este capítulo se resumen los experimentos realizados en el sistema en dos entornos diferentes y con cinco grupos de agentes diferentes. En primer lugar se describe la forma de evaluar dichos experimentos y la configuración de los entornos de pruebas. A continuación se evalúa cada una de las técnicas por separado en primer lugar para, en segundo lugar, se evalúa el algoritmo con todas sus características y finalmente se extraen conclusiones de la experimentación.

### 6.1. Método de evaluación

La forma de evaluar el rendimiento del sistema será observar el número de ciclos de media que tardan los agentes involucrados en el sistema en llegar al objetivo. Como la media puede ser muy variable de un episodio a otro, la medida elegida será la tendencia, es decir la media de las últimas 20 medias tomadas.

Así, la ilustración 46 contiene los ciclos que tarda cada uno de los dos jugadores presentes en cada episodio en resolver el problema. En verde podemos ver la media para todos los jugadores y en negro la citada tendencia.

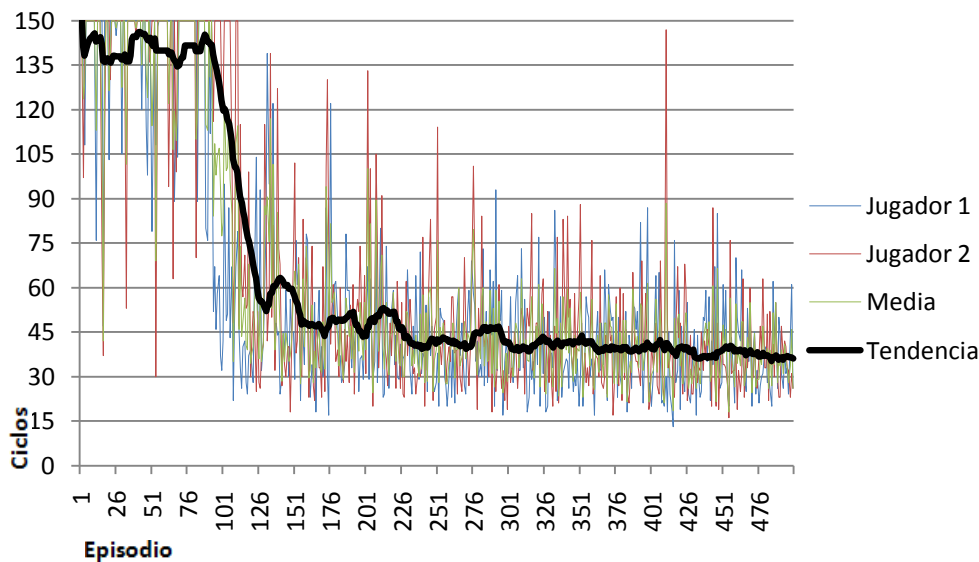
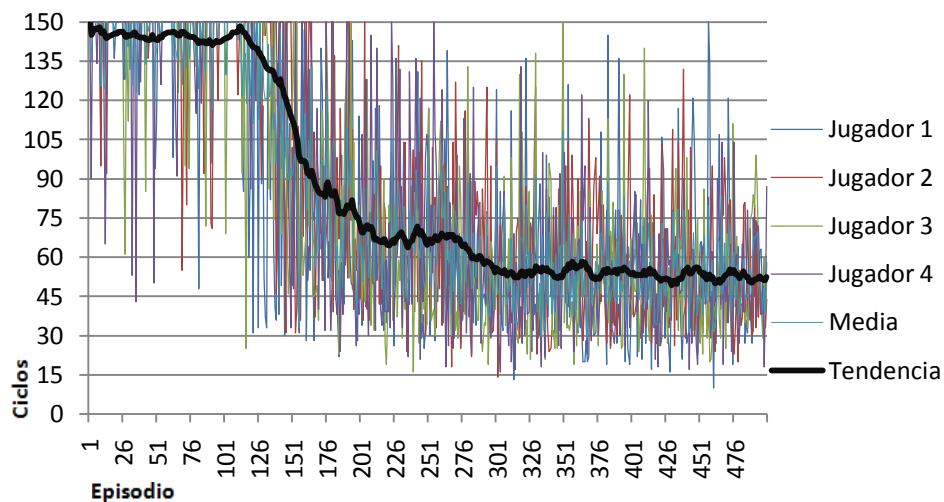


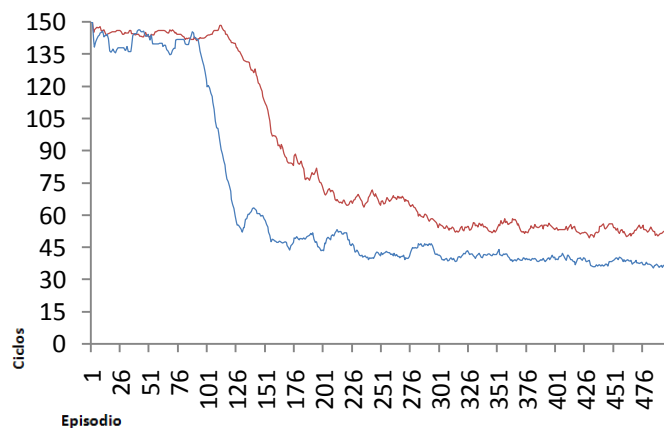
Ilustración 46 : Ejemplo de gráfico para dos jugadores

De igual forma, la ilustración 47 contiene el mismo experimento, pero esta vez para cuatro jugadores.



**Ilustración 47 : Ejemplo de gráfico para cuatro jugadores**

Por razones de visibilidad, las comparativas se efectuarán únicamente con dichas tendencias. Por ejemplo, para comparar los dos anteriores experimentos, se obtendrán gráficas como la presente en la ilustración 48:



**Ilustración 48 : Ejemplo de comparativa entre dos procesos de aprendizaje**

## 6.2. Entornos de prueba

En este apartado se explican los dos entornos de pruebas sobre los que se ha realizado el plan de experimentos.

### 6.2.1. Entorno Puerta

Simula la situación en la que todos los agentes, pertenecientes al mismo equipo, han de salir de la habitación por una puerta por la que sólo cabe uno de ellos.

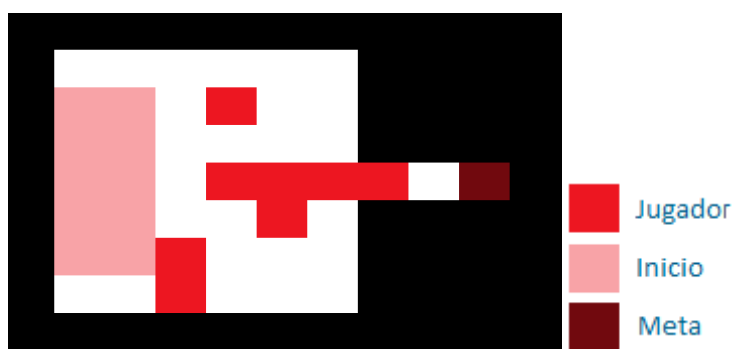


Ilustración 49 : Entorno puerta

Los experimentos en este entorno se harán sobre 1, 2, 4, 6 y 10 jugadores.

### 6.2.2. Entorno Pasillo:

Simula la situación en la que los agentes, repartidos en dos equipos diferentes, se encuentran en una habitación y tienen que alcanzar la habitación contigua atravesando un pasillo por el que sólo caben dos de ellos. Los equipos estarán en habitaciones contrapuestas, por lo que se tendrán que cruzar en el pasillo.

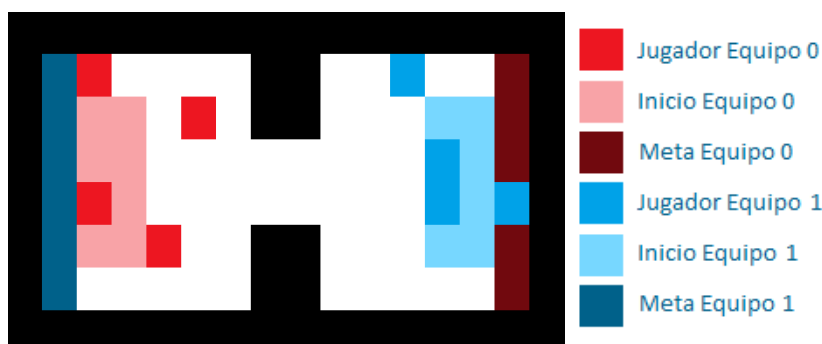


Ilustración 50 : Entorno pasillo

Los experimentos en este entorno se harán sobre 1, 2, 4, 8 y 16 jugadores.

### **6.3. Configuración de los experimentos**

Para cada uno de los entornos descritos, los experimentos seguirán el siguiente esquema:

#### **6.3.1. Caso base**

Se ejecutarán varios procesos de aprendizaje con igual configuración y distinto número de agentes ejecutando procesos de aprendizaje con el objetivo de tener una base sobre la que evaluar las distintas técnicas. El proceso de aprendizaje consiste en el algoritmo Q-Learning monoagente.

#### **6.3.2. Caso base con conocimiento previo del entorno**

Al entrenamiento en entorno compartido del caso base se le añade un proceso de entrenamiento en entorno exclusivo para que los agentes puedan acceder a un conocimiento previo del entorno en el que se encuentran. El objetivo de dicho conocimiento previo es acelerar la etapa de exploración de los procesos de aprendizaje en entorno exclusivo.

#### **6.3.3. Caso base con retardo de algunos procesos de aprendizaje**

Se modifica el entrenamiento en entorno compartido del caso base, de forma que al principio entrará un solo agente al entorno para que después, progresivamente, vayan uniéndose el resto. El objetivo de este retardo es reducir las interferencias de las etapas de exploración de los diferentes agentes entre sí.

#### **6.3.4. Caso base con política $\epsilon$ – greedy variable**

Se modifica la política de decisión de la acción a ejecutar en cada ciclo, añadiendo una política que modificará las probabilidades de elegir un movimiento aleatorio o uno aprendido según avance el proceso de aprendizaje. El objetivo es mejorar el equilibrio entre exploración y explotación.

#### **6.3.5. Caso base con conocimiento compartido.**

Al entrenamiento en entorno compartido del caso base se le añade la opción de que los agentes del mismo equipo puedan intercambiar el conocimiento que van adquiriendo.

#### **6.3.6. Algoritmo completo.**

Incluirá todas las técnicas anteriores para evaluar su funcionamiento conjunto con diferentes configuraciones de agentes.



## 6.4. Experimentos en el entorno Puerta

A continuación se detallan los experimentos acometidos sobre el entorno Puerta.

### Caso base en el entorno puerta

Como caso base se ejecutan tantos agentes como la prueba requiere. Cada agente tiene su propio proceso de Q-Learning configurado como sigue.

CASO BASE - PUERTA											
Parámetros Q-Learning											
<b>Episodios</b>	300	<b>Ciclos</b>	150	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	0.4	<b>Refuerzos negativos</b>	SI
Técnicas implementadas usadas											
<b>Previo</b>	NO	<b>Retardo</b>	NO	<b>Conocimiento compartido</b>				NO	<b>Núm. Agentes</b>	1,2,4,6 y 10	

Tabla 6 : Configuración para la ejecución del caso base en el entorno puerta

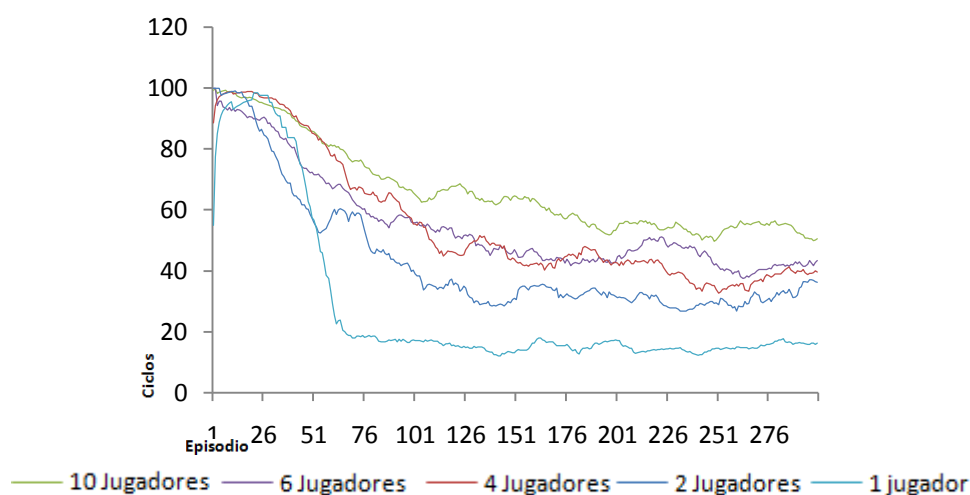


Ilustración 51 : Comparativa las distintas configuraciones de agentes para el caso base en el entorno puerta

En la ilustración 51 podemos observar los cinco procesos de aprendizaje que serán tomados como caso base. El resultado es el esperado, aumentando el número de ciclos conforme aumenta el número de agentes presente en el sistema. Además, el número de episodios necesarios para alcanzar una cierta estabilidad también aumenta con el número de agentes presentes.

### **Experimento 1: Caso base con conocimiento previo del entorno en el entorno puerta**

En este conjunto de experimentos se añadirá al caso base conocimiento previo del entorno, obtenido mediante procesos de aprendizaje ejecutados en un entorno vacío.

EXPERIMENTO 1											
Parámetros Q-Learning											
Episodios	300	Ciclos	100	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	0.4	Refuerzos negativos	SI
Técnicas implementadas usadas											
Previo	SI	Retardo	NO	Conocimiento compartido				NO	Núm. Agentes	1,2,4,6 y 10	
Parámetros Q-Learning para conocimiento previo											
Episodios	200	Ciclos	100	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	0.4	Refuerzos negativos	SI

Tabla 7 : Configuración del experimento 1

### Comparativa frente al caso base

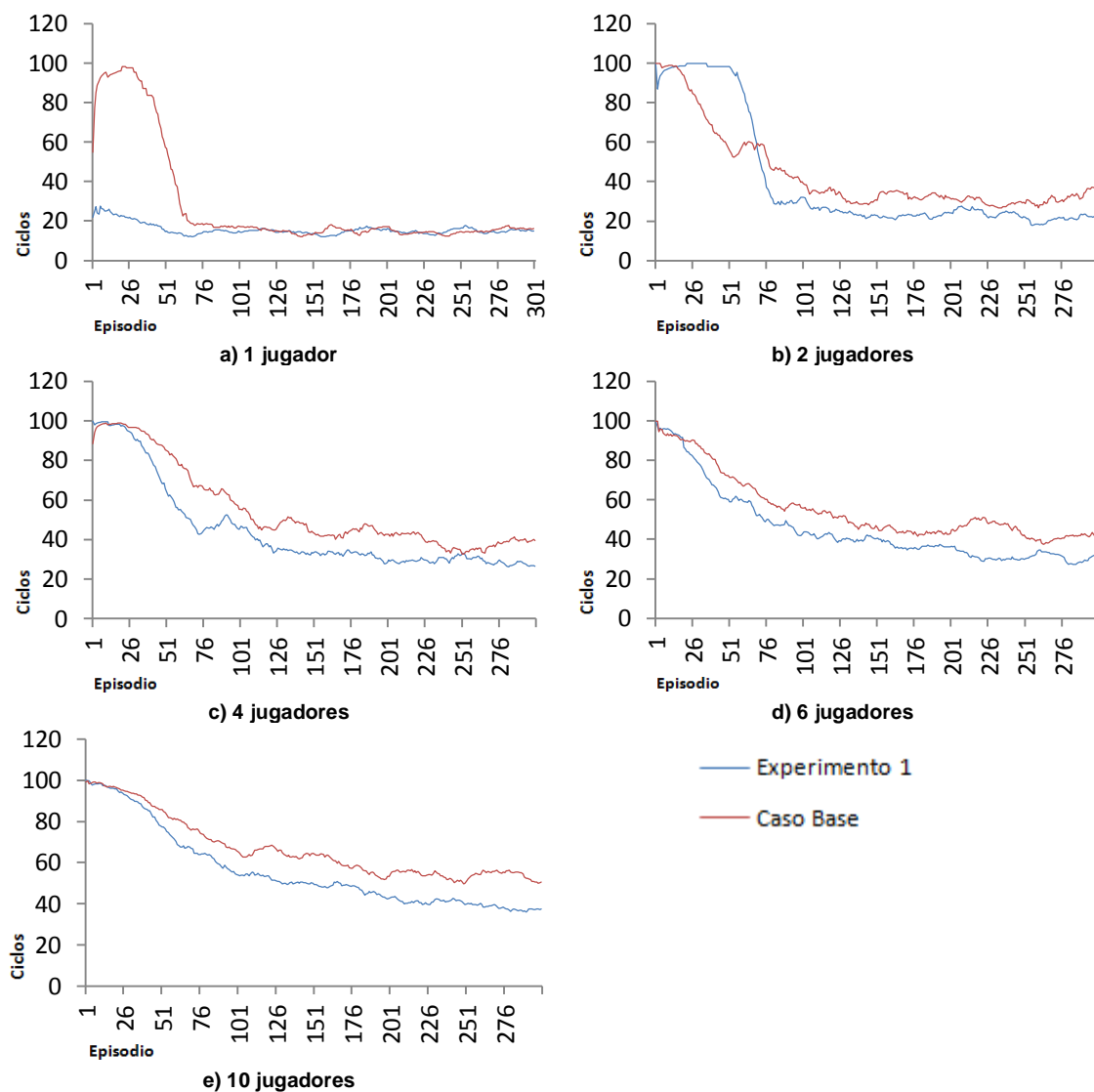


Ilustración 52: Comparativa entre el experimento 1 y el caso base

En este caso, la aceleración del proceso de aprendizaje se puede observar en los casos en los que menos agentes interactúan en el entorno. Aunque los agentes conocen el camino a la meta no han aprendido a esquivarse entre ellos y, por tanto, se interrumpen en dicho camino haciendo que la técnica pierda rendimiento cuando crece el número de agentes.

No obstante, comparando las etapas finales de los cinco experimentos podemos observar que cuando hay más agentes, la mejora debida a esta técnica se incrementa.

## **Experimento 2: Caso base con retardo de los procesos de aprendizaje en el entorno puerta**

En este conjunto de experimentos se retardará el proceso de aprendizaje de los agentes, entrando a ejecutar un agente nuevo cada diez episodios.

<b>EXPERIMENTO 2</b>											
<b>Parámetros Q-Learning</b>											
<b>Episodios</b>	300	<b>Ciclos</b>	100	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	0.4	<b>Refuerzos negativos</b>	SI
<b>Técnicas implementadas usadas</b>											
<b>Previo</b>	NO	<b>Retardo</b>	10	<b>Conocimiento compartido</b>				NO	<b>Núm. Agentes</b>	1,2,4,6 y 10	

Tabla 8 : Configuración del experimento 2

Siguiendo la presente configuración, en el experimento estarán todos los agentes en el entorno:

- 1 jugador: Presente desde el primer episodio.
- 2 jugadores: Presentes desde el episodio 10.
- 4 jugadores: Presentes desde el episodio 30.
- 6 jugadores: Presentes desde el episodio 60.
- 10 jugadores: Presentes desde el episodio 90.

En los gráficos se indica con una línea vertical negra el primer episodio en el que todos los agentes están presentes en el sistema.

### Comparativa frente al caso base

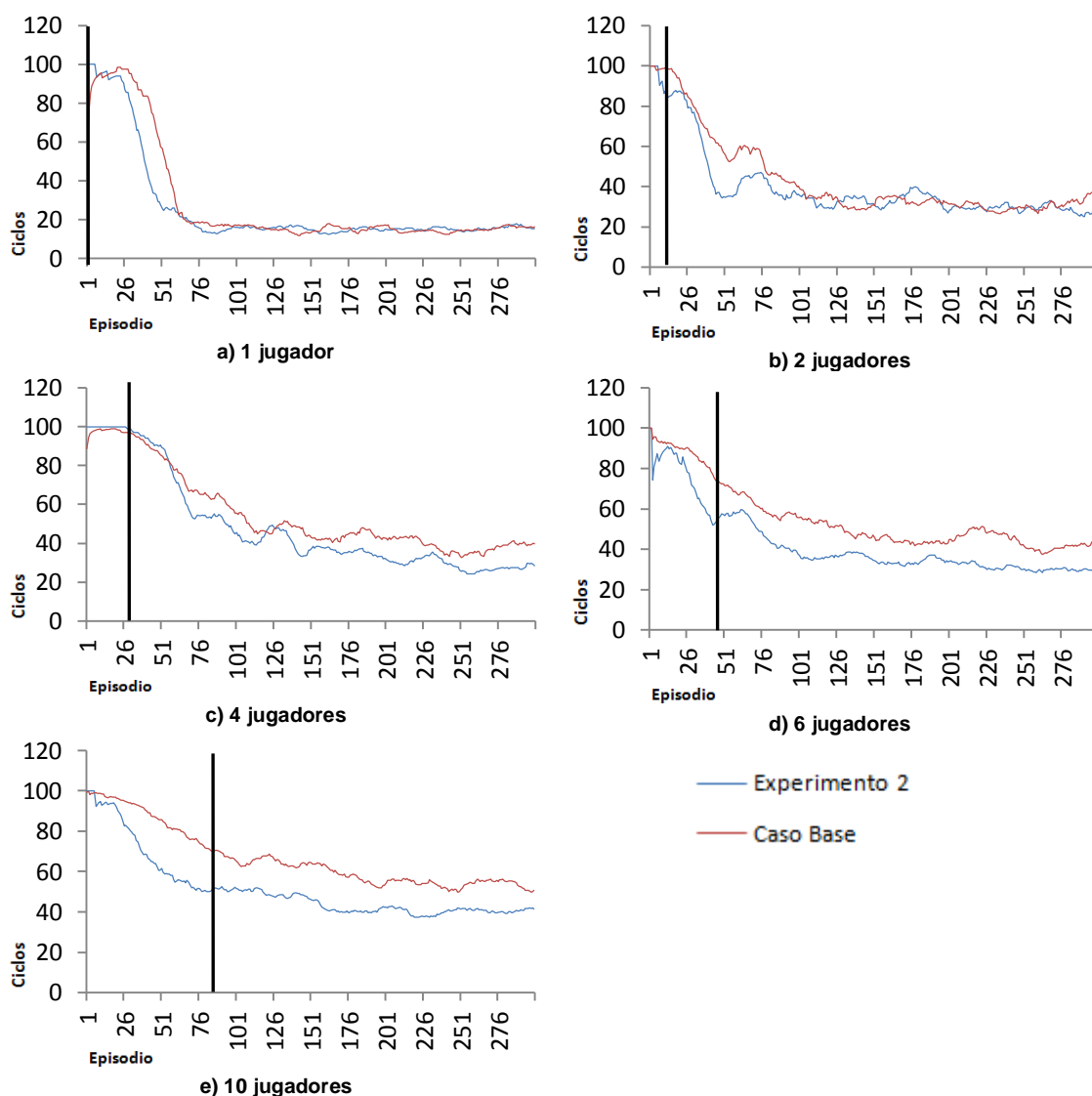


Ilustración 53 : Comparativa entre el experimento 2 y el caso base

Comparando los resultados de los experimentos frente al caso base se observa que apenas hay mejora cuando se trata de pocos agentes. Pero cuando el número de agentes se incrementa, la técnica aporta una mejora significativa al proceso de aprendizaje. Ello es debido a que, al aparecer todos los agentes a la vez en el sistema hay demasiadas interrupciones. Añadiendo poco a poco los agentes al sistema las interrupciones serán menores y, por tanto, los agentes podrán ir adaptándose.

### Experimento 3: Caso base con política e-greedy variable en el entorno puerta

En este conjunto de experimentos se ejecutará el proceso de aprendizaje del caso base, añadiéndole la política de decisión de acción e-greedy variable configurada según la tabla 9.

EXPERIMENTO 3											
Parámetros Q-Learning											
Episodios	300	Ciclos	100	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	variable	Refuerzos negativos	SI
Técnicas implementadas usadas											
Previo	NO	Retardo	NO	Conocimiento compartido				NO	Núm. Agentes	1,2,4,6 y 10	
Parámetros política de decisión de la acción variable.											
$\epsilon$ inicial (mín,máx)			0.6 (0.1,0.4)			Incremento $\epsilon$				-0.002	
greedy exclusivo inicial(mín,máx)			0.0 (0,0.0)			Incremento greedy exclusivo				0	
greedy compartido inicial (mín,máx)			0.4 (0,1)			Incremento greedy compartido				0.002	

Tabla 9 : Configuración del experimento 3

Con esta configuración, si todos los agentes llegarán en todos los episodios a la meta, la política de decisión de la acción será 100% greedy. Como dicha la premisa de que todos los agentes lleguen en todos los episodios a la meta, lo habitual es que se ejecuten políticas en torno al 95% greedy.

### Comparativa frente al caso base

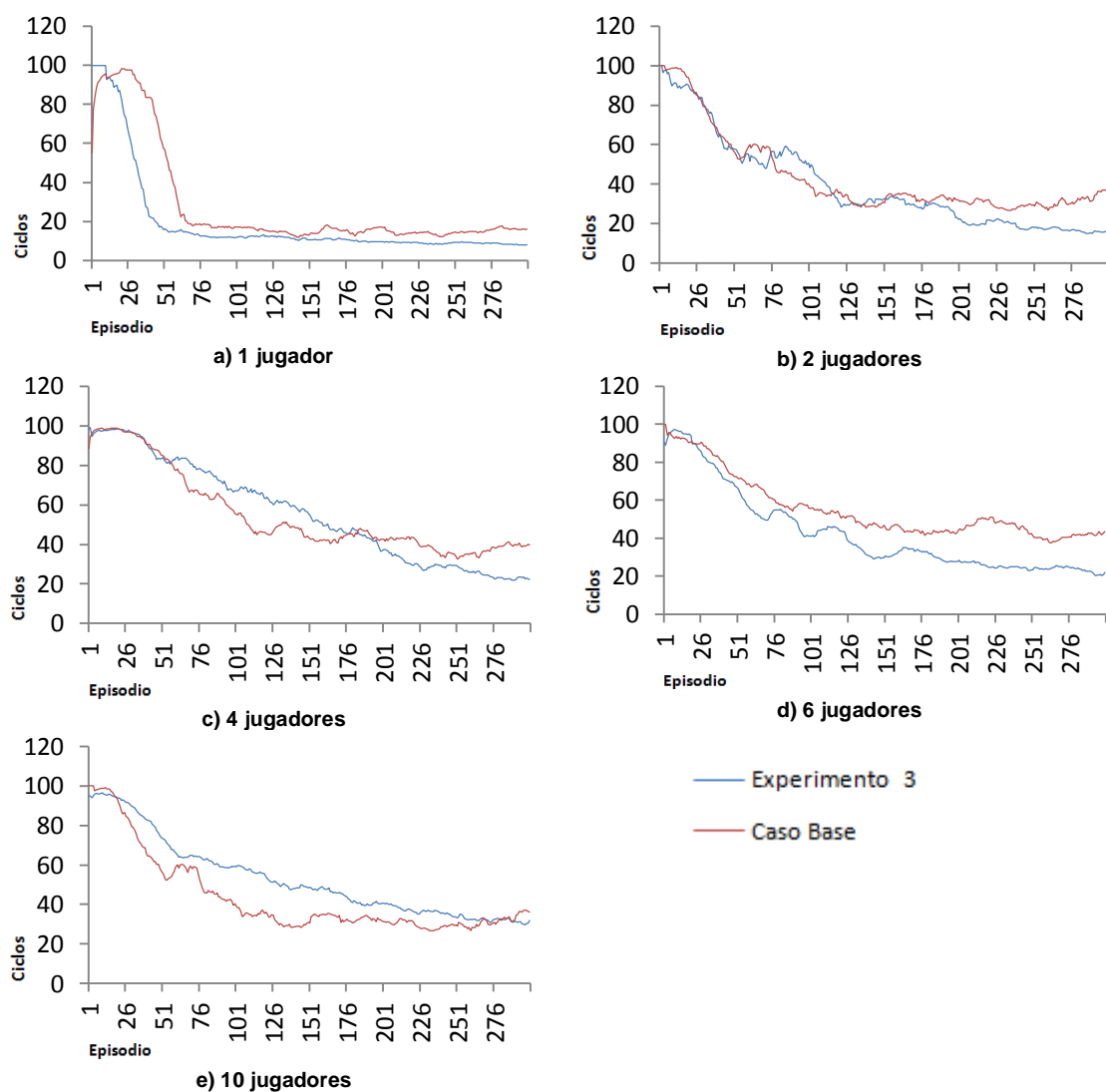


Ilustración 54: Comparativa entre el experimento 3 y el caso base

En comparación con el caso base, la mejora es menor que con las otras técnicas. Dejando de lado el caso de 6 jugadores (d), producto del azar, la técnica sólo mejora el caso base en la fase final del experimento y no es realmente significativa. Para obtener todo el beneficio de esta técnica hay que combinarla con otras técnicas como obtener un conocimiento previo del entorno.

#### **Experimento 4: Caso base con conocimiento compartido en el entorno puerta**

En este conjunto de experimentos se añade al caso base la posibilidad de que los agentes compartan los conocimientos adquiridos durante sus procesos de aprendizaje.

EXPERIMENTO 4											
Parámetros Q-Learning											
Episodios	300	Ciclos	100	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	0.4	Refuerzos negativos	SI
Técnicas implementadas usadas											
Previo	NO	Retardo	NO	Conocimiento compartido				SI	Núm. Agentes	1,2,4,6 y 10	
Parámetros conocimiento compartido											
Episodios		25				Ciclos		80			

Tabla 10 : Configuración del experimento 4

De esta forma todos los agentes enviarán la política que llevan aprendida cada vez que acaben un episodio de aprendizaje para que sea procesada por el gestor de conocimiento. Además, cada vez que un agente alcance la meta en más de 80 ciclos o no la alcance requerirá conocimiento del gestor de conocimiento. En cualquier caso, el agente no pasará más de 25 episodios sin requerir dicho conocimiento.



### Comparativa frente al caso base

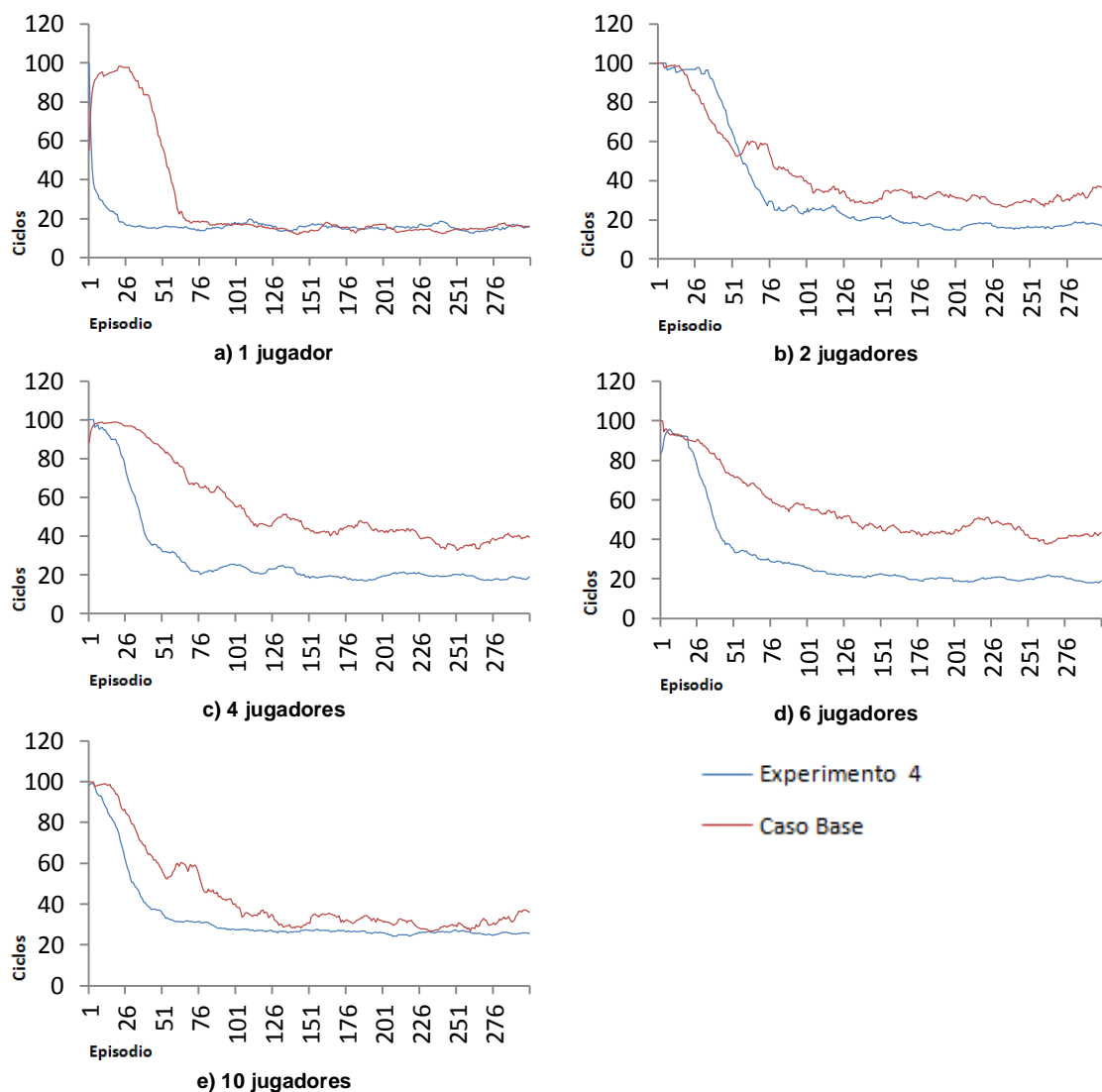


Ilustración 55: Comparativa entre el experimento 4 y el caso base

Como se puede observar compartir el conocimiento adquirido aporta beneficios en el proceso de aprendizaje. No obstante, la mejoría es mucho mayor para 4 (c) y 6 (d) jugadores que para 10 jugadores (e). Destacar de nuevo la estabilidad del aprendizaje frente al caso base, las variaciones entre episodios son mucho menores que con el caso base.

### Experimento 5: Algoritmo completo en el entorno puerta

En este experimento se ejecuta el algoritmo sobre tantos agentes como la prueba requiere. El algoritmo usa todas sus características configuradas según la tabla 11.

EXPERIMENTO 5											
Parámetros Q-Learning											
Episodios	300	Ciclos	100	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	Variable	Refuerzos negativos	SI
Técnicas implementadas usadas											
Previo	SI	Retardo	10	Conocimiento compartido				SI	Núm. Agentes	1,2,4,6 y 10	
Parámetros Q-Learning para conocimiento previo											
Episodios	200	Ciclos	100	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	0.4	Refuerzos negativos	SI
Parámetros conocimiento compartido											
Episodios		25				Ciclos		80			
Parámetros política de decisión de la acción variable.											
$\epsilon$ inicial (mín,máx)			0.4 (0.1,0.4)			Incremento $\epsilon$				-0.001	
greedy exclusivo inicial(mín,máx)			0.6 (0,0.6)			Incremento greedy exclusivo				-0.002	
greedy compartido inicial (mín,máx)			0 (0,1)			Incremento greedy compartido				0.003	

Tabla 11 : Configuración del experimento 5

Al usar la técnica de retardo de algunos procesos de aprendizaje, se incluye una línea negra que indica el primer episodio en el que todos los jugadores están ejecutando.

### Comparativa frente al caso base

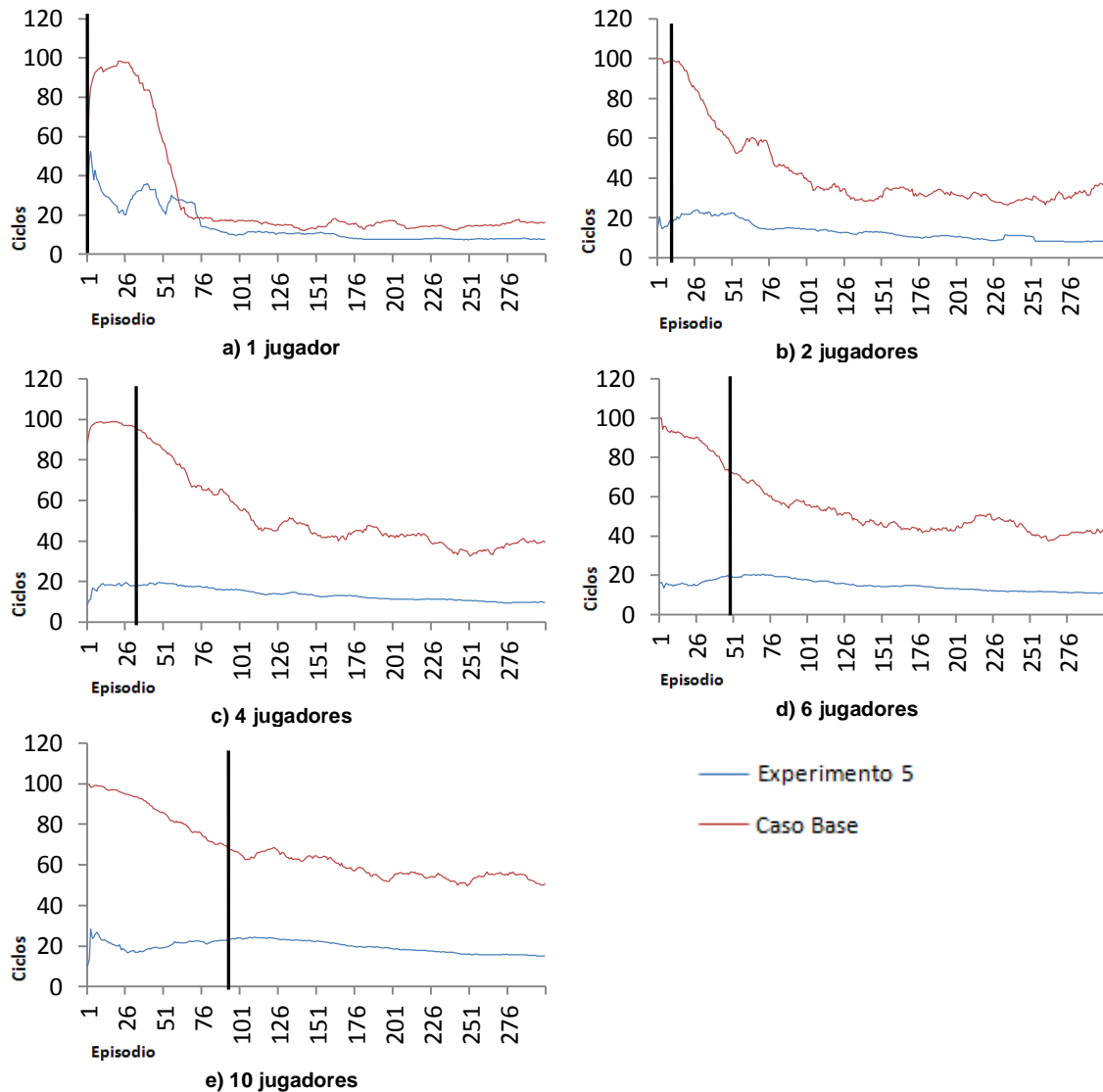


Ilustración 56 : Comparativa entre el experimento 5 y el caso base

Los resultados muestran que el algoritmo mejora los resultados obtenidos por el caso base. La mejora apenas es significativa cuando es un único agente el presente en el sistema y es principalmente debida al conocimiento previo del entorno y al método  $\epsilon - greedy$  variable. La mejora se hace más significativa cuando el número de agentes en el sistema incrementa, puesto que la cantidad de conocimiento compartido aumenta.

Sin embargo, los resultados obtenidos para el algoritmo para un único agente (a) son muy superiores a los obtenidos para 10 jugadores (e). Ello es síntoma de que el algoritmo todavía tiene mucho margen de mejora.

## 6.5. Experimentos en el entorno Pasillo

A continuación se detallan los experimentos acometidos sobre el entorno Pasillo.

### Caso base

Como caso base se ejecutan nuevamente tantos agentes como la prueba requiere. Cada agente tiene su propio proceso de Q-Learning configurado como sigue.

<b>CASO BASE - PASILLO</b>											
<b>Parámetros Q-Learning</b>											
<b>Episodios</b>	500	<b>Ciclos</b>	150	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	0.6	<b>Refuerzos negativos</b>	SI
<b>Técnicas implementadas usadas</b>											
<b>Previo</b>	NO	<b>Retardo</b>	NO	<b>Conocimiento compartido</b>				NO	<b>Núm. Agentes</b>	1,2,4,8 y 16	

Tabla 12 : Configuración para la ejecución del caso base en el entorno pasillo

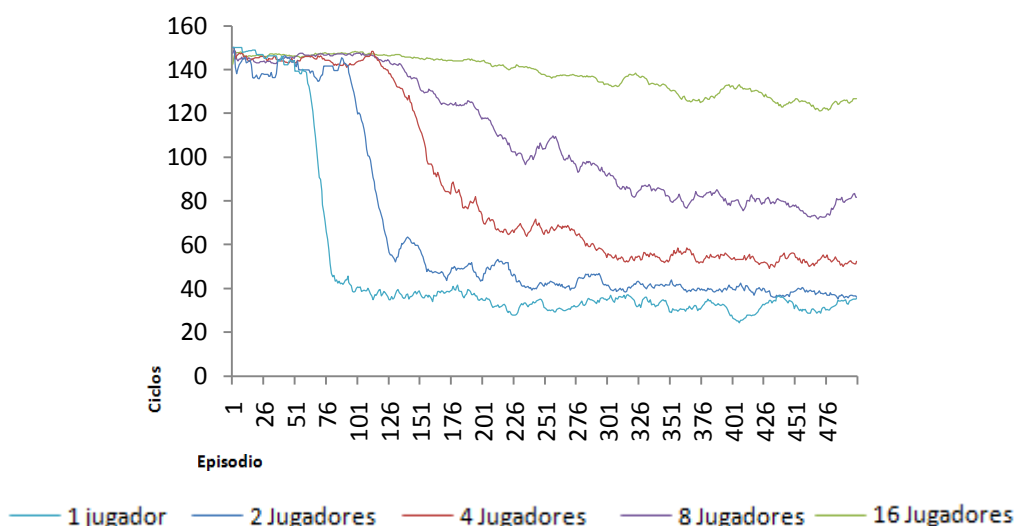


Ilustración 57 : Comparativa las distintas configuraciones de agentes para el caso base en el entorno pasillo

En la ilustración 57 podemos observar los cinco procesos de aprendizaje que serán tomados como caso base. El resultado es nuevamente el esperado, aumentando el número de ciclos conforme aumenta el número de agentes presente en el sistema. De igual forma, el número de episodios necesarios para alcanzar una cierta estabilidad también aumenta con el número de agentes presentes.



### **Experimento 6: Caso base con conocimiento previo del entorno en un entorno pasillo**

En este conjunto de experimentos se añadirá al caso base conocimiento previo del entorno, obtenido mediante procesos de aprendizaje ejecutados en un entorno vacío.

EXPERIMENTO 6											
Parámetros Q-Learning											
Episodios	500	Ciclos	150	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	0.6	Refuerzos negativos	SI
Técnicas implementadas usadas											
Previo	SI	Retardo	NO	Conocimiento compartido				NO	Núm. Agentes	1,2,4,8 y 16	
Parámetros Q-Learning para conocimiento previo											
Episodios	300	Ciclos	150	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	0.6	Refuerzos negativos	SI

Tabla 13 : Configuración del experimento 6

### Comparativa frente al caso base

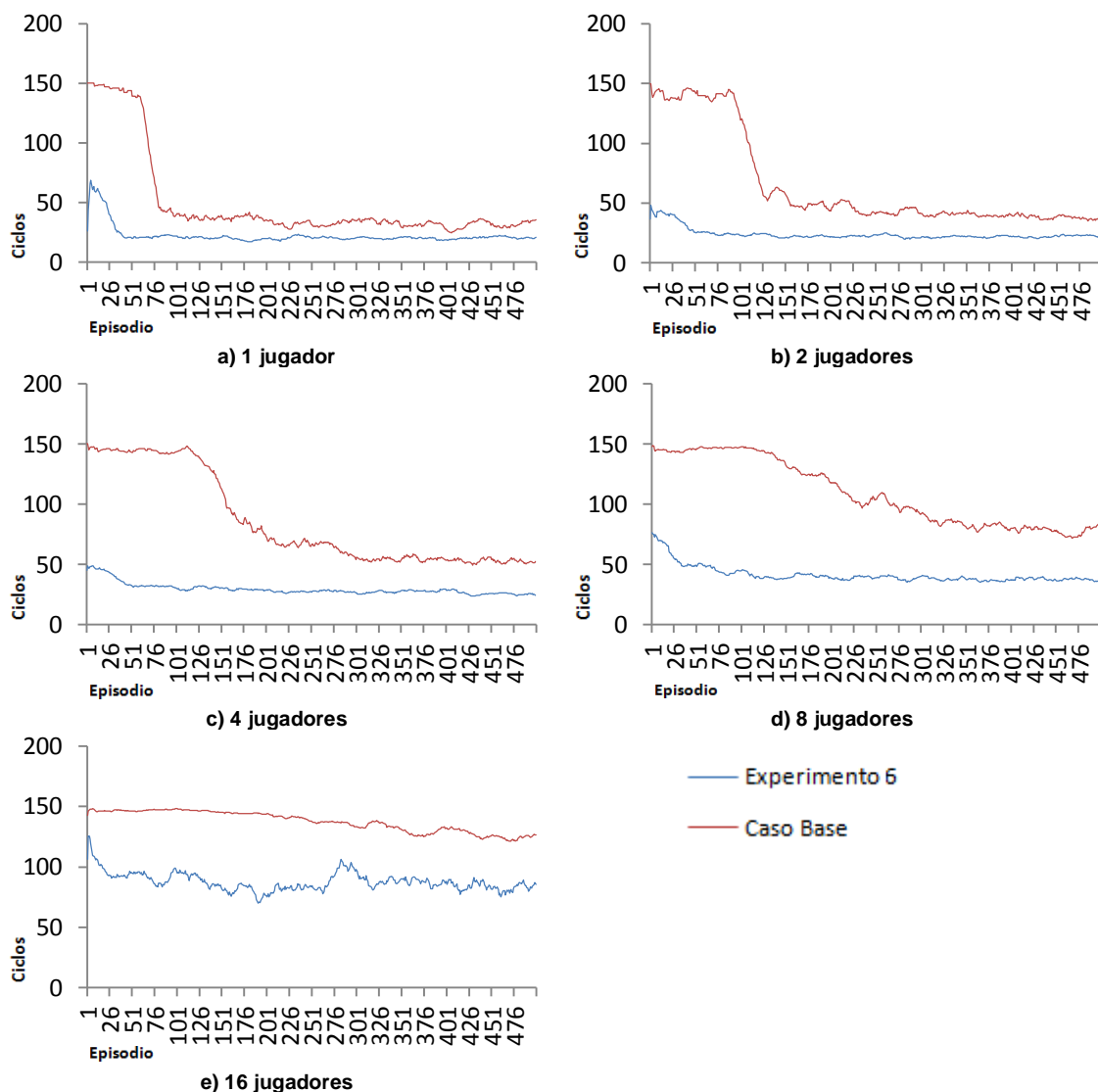


Ilustración 58 : Comparativa entre el experimento 6 y el caso base

De nuevo, se puede observar la aceleración del proceso de aprendizaje en los experimentos en los que menos agentes intervienen. Además, comparando con el caso base, podemos observar que la estabilidad del aprendizaje se alcanza mucho antes. Esta circunstancia además hace que el proceso obtenga mejores resultados puesto que sólo se tiene que ocupar de aprender a evitar interrupciones de otros agentes.

### **Experimento 7: Caso base con retardo de los procesos de aprendizaje en el entorno pasillo**

En este conjunto de experimentos se retardará el proceso de aprendizaje de los agentes, entrando a ejecutar un agente nuevo cada diez episodios.

<b>EXPERIMENTO 7</b>											
<b>Parámetros Q-Learning</b>											
<b>Episodios</b>	500	<b>Ciclos</b>	150	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	0.6	<b>Refuerzos negativos</b>	SI
<b>Técnicas implementadas usadas</b>											
<b>Previo</b>	NO	<b>Retardo</b>	10	<b>Conocimiento compartido</b>				NO	<b>Núm. Agentes</b>	1,2,4,8 y 16	

Tabla 14 : Configuración del experimento 7

Siguiendo la presente configuración, en el experimento estarán todos los agentes en el entorno:

- 1 jugador: Presente desde el primer episodio.
- 2 jugadores: Presentes desde el episodio 10.
- 4 jugadores: Presentes desde el episodio 30.
- 8 jugadores: Presentes desde el episodio 70.
- 16 jugadores: Presentes desde el episodio 150.

En los gráficos se indica con una línea vertical negra el primer episodio en el que todos los agentes están presentes en el sistema.



### Comparativa frente al caso base

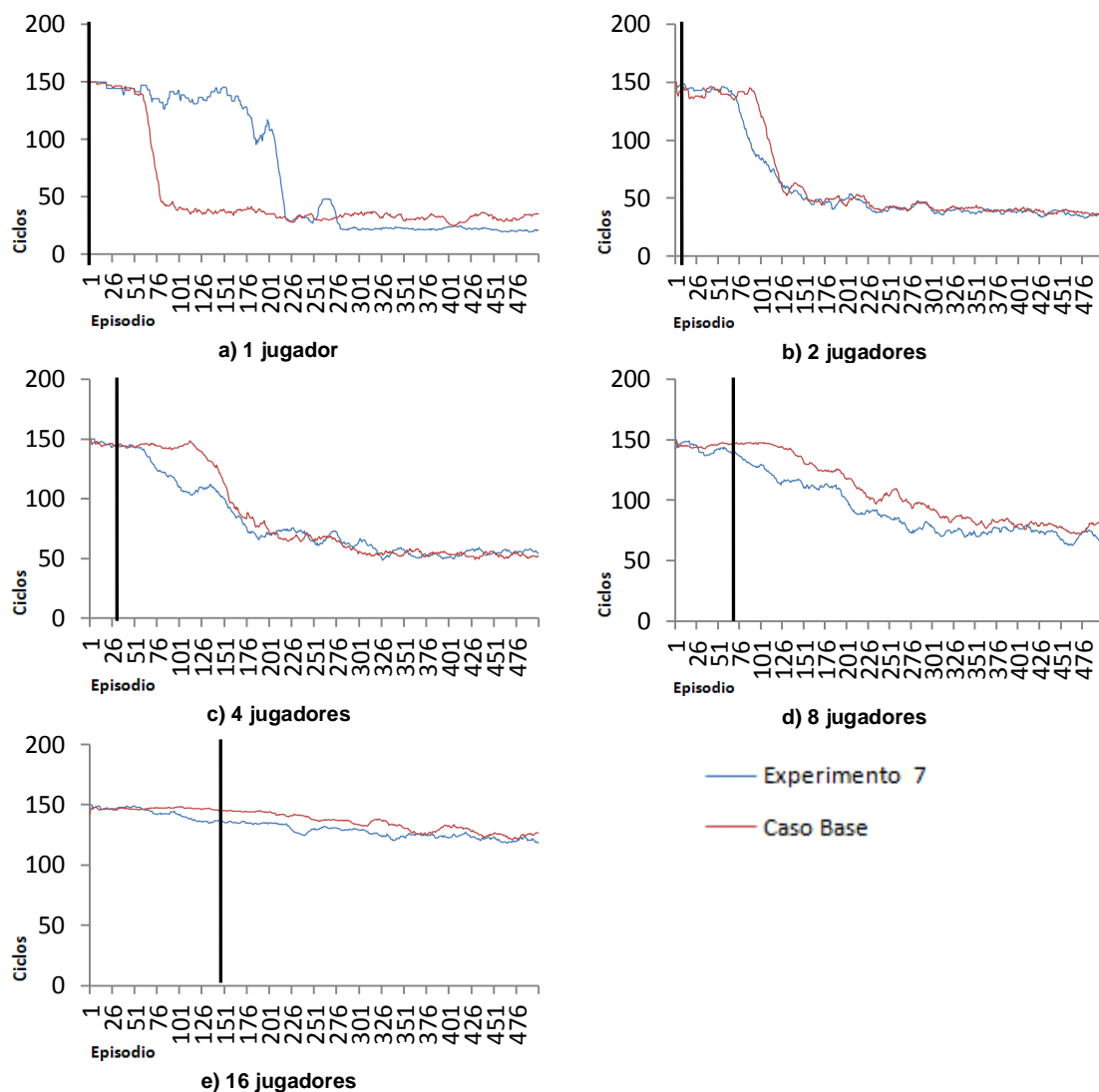


Ilustración 59 : Comparativa entre el experimento 7 y el caso base

Como se puede observar los resultados son muy similares a los del caso base tomado. Aunque existe una mejora, esta no es realmente muy significativa. Para usar esta técnica en entornos competitivos es necesaria combinarla con otras técnicas que aporten conocimiento al agente, puesto que no es útil retardar el aprendizaje de algunos agentes si los que se encuentran en el entorno no consiguen valerse de esa ventaja.

### Experimento 8: Caso base con política e-greedy variable en el entorno pasillo

En este conjunto de experimentos se ejecutará el proceso de aprendizaje del caso base, añadiéndole la política de decisión de acción e-greedy variable configurada según la tabla 15.

EXPERIMENTO 8											
Parámetros Q-Learning											
Episodios	500	Ciclos	150	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	Variable	Refuerzos negativos	SI
Técnicas implementadas usadas											
Previo	NO	Retardo	NO	Conocimiento compartido				NO	Núm. Agentes	1,2,4,8 y 16	
Parámetros política de decisión de la acción variable.											
$\epsilon$ inicial (mín,máx)			0.6 (0.1,0.6)			Incremento $\epsilon$				-0.0012	
greedy exclusivo inicial(mín,máx)			0.0 (0,0)			Incremento greedy exclusivo				0	
greedy compartido inicial (mín,máx)			0.4 (0,1)			Incremento greedy compartido				0.0012	

Tabla 15 : Configuración del experimento 8

Con esta configuración, si todos los agentes llegarán en todos los episodios a la meta, la política de decisión de la acción será 100% greedy. Como dicha la premisa de que todos los agentes lleguen en todos los episodios a la meta, lo habitual es que se ejecuten políticas en torno al 95% greedy.

### Comparativa frente al caso base

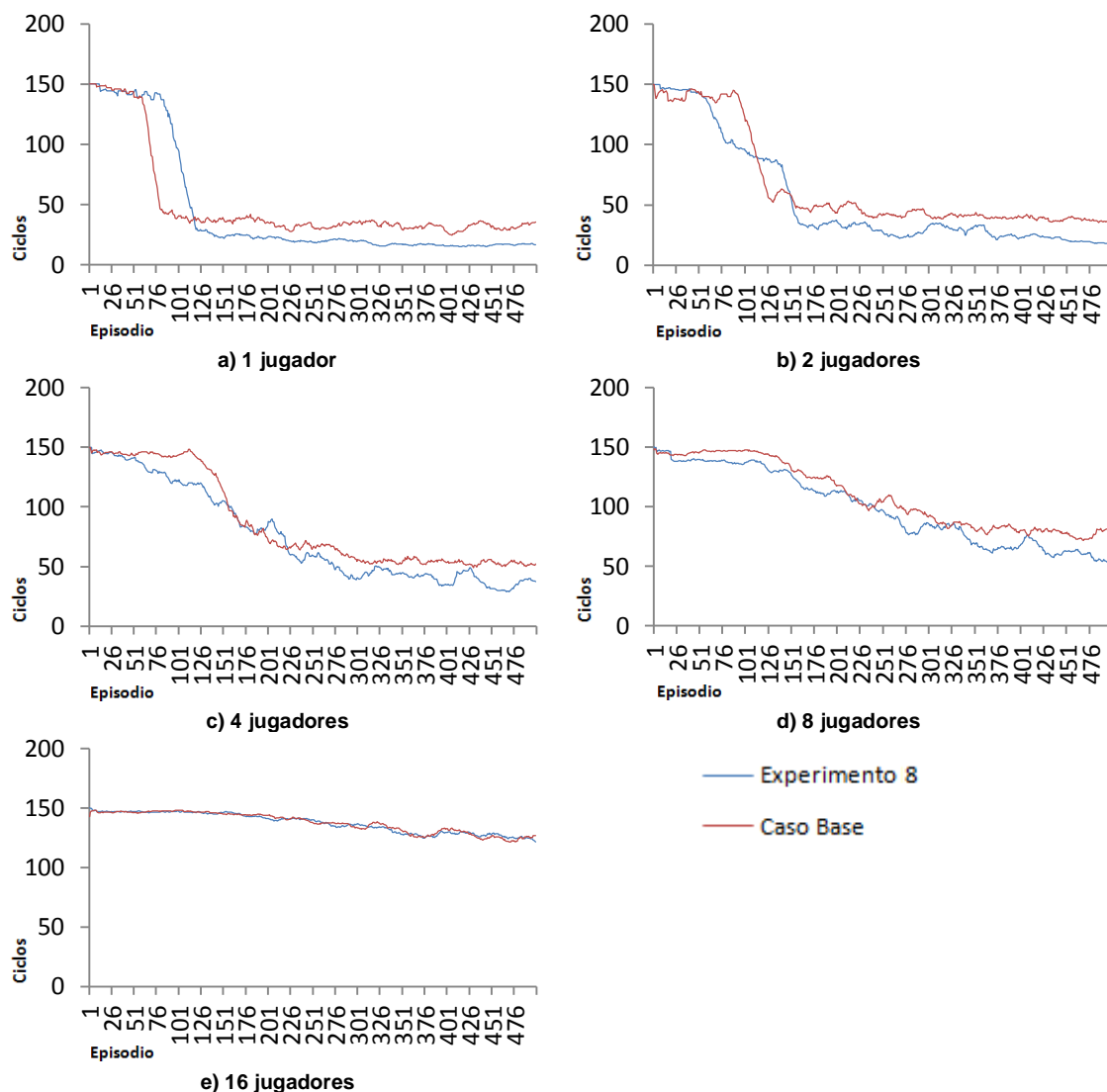


Ilustración 60: Comparativa entre el experimento 8 y el caso base

De nuevo vemos que, en un entorno competitivo, la técnica e-greedy no consigue resultados significativos por si misma cuando el número de agentes se incrementa.

### Experimento 9: Caso base con conocimiento compartido

En este conjunto de experimentos se añade al caso base la posibilidad de que los agentes compartan los conocimientos adquiridos durante sus procesos de aprendizaje.

EXPERIMENTO 9											
Parámetros Q-Learning											
Episodios	500	Ciclos	150	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	0.6	Refuerzos negativos	SI
Técnicas implementadas usadas											
Previo	NO	Retardo	NO	Conocimiento compartido				SI	Núm. Agentes	1,2,4,8 y 16	
Parámetros conocimiento compartido											
Episodios		25				Ciclos		145			

Tabla 16 : Configuración del experimento 9

De esta forma todos los agentes enviarán la política que llevan aprendida cada vez que acaben un episodio de aprendizaje para que sea procesada por el gestor de conocimiento. Además, cada vez que un agente alcance la meta en más de 145 ciclos o no la alcance requerirá conocimiento del gestor de conocimiento. En cualquier caso, el agente no pasará más de 25 episodios sin requerir dicho conocimiento.

### Comparativa frente al caso base

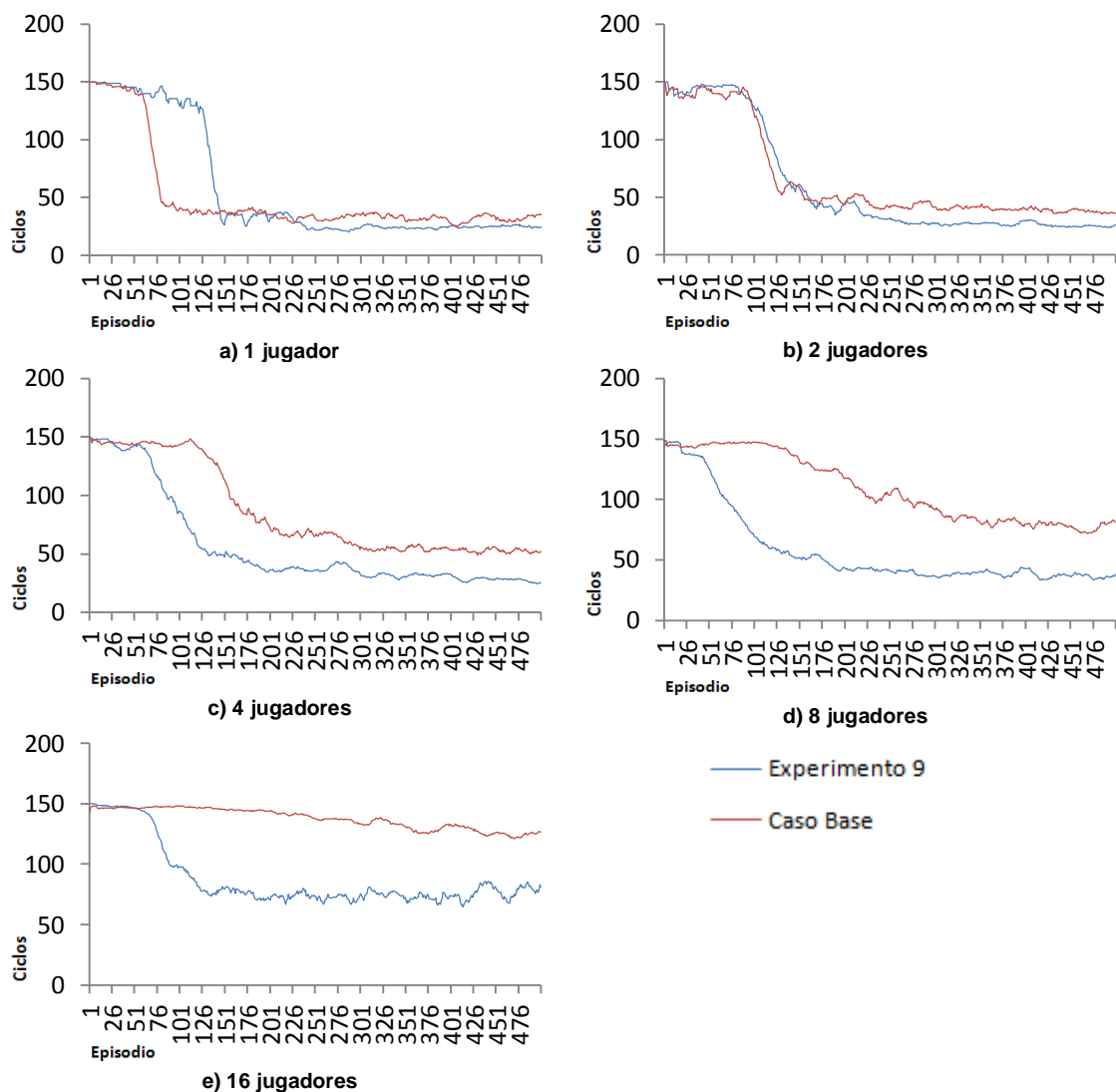


Ilustración 61: Comparativa entre el experimento 9 y el caso base

En la comparativa se puede observar la enorme mejora que aporta esta técnica. Además, para este entorno competitivo, la mejoría es más notable cuando aumenta el número de agentes implicados a diferencia del entorno anterior.

### Experimento 10: Algoritmo completo en el entorno pasillo

En este conjunto de experimentos se ejecuta el algoritmo sobre tantos agentes como la prueba requiere. El algoritmo usa todas sus características configuradas según la tabla 9.

EXPERIMENTO 10											
Parámetros Q-Learning											
Episodios	500	Ciclos	150	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	Variable	Refuerzos negativos	SI
Técnicas implementadas usadas											
Previo	SI	Retardo	10	Conocimiento compartido				SI	Núm. Agentes	1,2,4,8 y 16	
Parámetros Q-Learning para conocimiento previo											
Episodios	300	Ciclos	150	$\alpha$	0.95	$\gamma$	0.95	$\epsilon$	0.6	Refuerzos negativos	SI
Parámetros conocimiento compartido											
Episodios		25				Ciclos		145			
Parámetros política de decisión de la acción variable.											
$\epsilon$ inicial (mín,máx)			0.4 (0.1,0.4)			Incremento $\epsilon$				-0.0006	
greedy exclusivo inicial(mín,máx)			0.6 (0,0.6)			Incremento greedy exclusivo				-0.0012	
greedy compartido inicial (mín,máx)			0 (0,1)			Incremento greedy compartido				0.0018	

Tabla 17 : Configuración del experimento 10

Al usar la técnica de retardo de algunos procesos de aprendizaje, se incluye una línea negra que indica el primer episodio en el que todos los jugadores están ejecutando.

### Comparativa frente al caso base

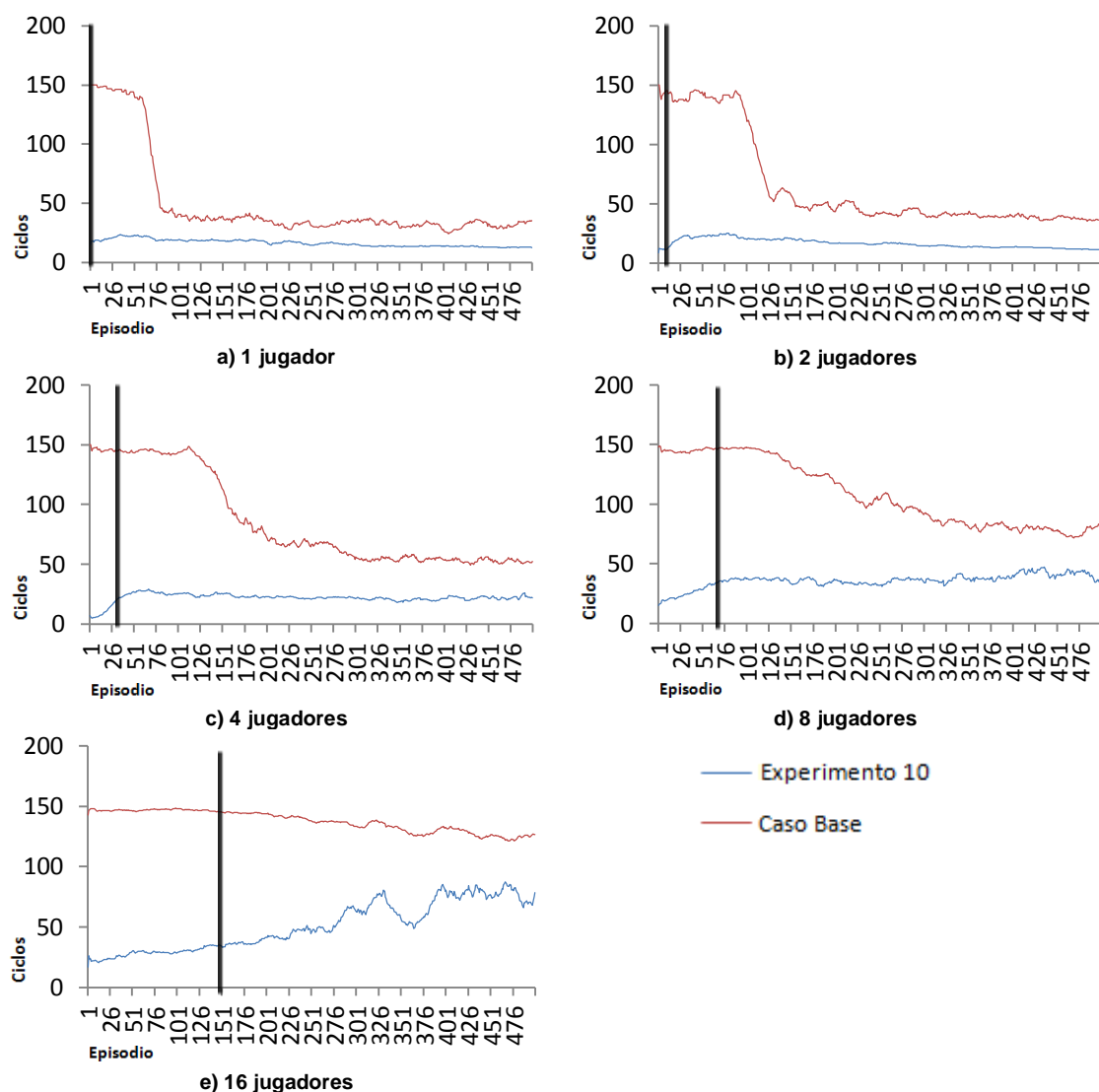


Ilustración 62 : Comparativa entre el experimento 10 y el caso base

De nuevo vemos como el algoritmo mejora los resultados obtenidos por el caso base. En este caso el número de agentes afecta muy negativamente al proceso de aprendizaje, debido a la gran cantidad de bloqueos entre los propios agentes. Ello se pone de manifiesto en el final del experimento con 16 agentes, en el que el sistema llega a desaprender, puesto que la política seguida en torno al episodio 150 (primer episodio en el que todos los agentes están presentes) da mejores resultados que la seguida al final del proceso.

## 6.6. Conclusiones de la experimentación

En general, el algoritmo propuesto mejora los resultados obtenidos por el caso base. Particularmente para cada técnica:

- **Conocimiento previo del entorno:** Su aportación se nota especialmente en la fase inicial del proceso de aprendizaje. Los agentes ya saben dónde está la meta y cómo llegar a ella, por lo que sólo deben aprender a interaccionar con otros agentes. En resumen, esta técnica **facilita en gran medida la tarea de exploración** del entorno.
- **Retardo de los procesos de aprendizaje:** Facilita de nuevo **la tarea de exploración del entorno**. A diferencia del uso de conocimiento previo del entorno, aporta sus facilidades en las interacciones entre los agentes. El hecho de que al principio haya pocos agentes con los que interaccionar hace que sea más fácil. Así, según van adquiriendo conocimiento, pueden irse enfrentando a los problemas de mayor envergadura que aparecen con todos los agentes presentes. Por otro lado, si este parámetro está mal ajustado, es posible que los primeros agentes en ejecutar han aprendido políticas muy especializadas para pocos agentes y no puedan adaptarse a un número muy superior de agentes.
- **e-greedy variable:** Gracias a esta técnica el algoritmo puede equilibrar el **equilibrio entre la exploración** del entorno y la **explotación** del conocimiento adquirido. Pero esta técnica no puede llegar a permitir ejecutar políticas 100% greedy, puesto que estas políticas no funcionan correctamente. Los agentes aprenden políticas rígidas para entornos cambiantes, por lo que es habitual que al intentar hacer todo lo mismo se bloqueen. Es necesario un mínimo porcentaje de aleatoriedad (en torno al 5%) para que las políticas aprendidas funcionen correctamente.
- **Conocimiento compartido:** Su uso potencia **tanto la exploración** del entorno como la **explotación** del conocimiento. Por un lado, puesto que los agentes pueden aprender de la exploración efectuada por otros. Por otro, los agentes pueden obtener mejores políticas que explotar durante el aprendizaje. Sin embargo, si los agentes comparten demasiado conocimiento se limita en gran medida la exploración del entorno, puesto que todos los agentes se mueven tomando más o menos las mismas decisiones. Además, en determinados entornos puede no ser conveniente que todos los agentes exploten de la misma forma el conocimiento.

A la hora de conjugar todas las técnicas, en el entorno Puerta el algoritmo implementado mejora con mucho el caso base y el resultado parcial de cada una de las técnicas.



Sin embargo, ese no es el resultado que se da para el entorno Pasillo. En dicho entorno, aunque el algoritmo mejora al caso base y a cada una de las técnicas, lo cierto es que **llega a desaprender**. Llama la atención como en el experimento 10 para 16 jugadores, el algoritmo obtiene mejor resultado con la política ejecutada en el episodio 150 con aproximadamente un 37% de aleatoriedad que al final del experimento, con un 5% de aleatoriedad.



## 7. Conclusiones y trabajos futuros

En esta sección se extraen conclusiones acerca del trabajo realizado y de los resultados obtenidos en la fase de pruebas. Además se proponen nuevas líneas de trabajo que se pueden tomar a partir del trabajo realizado.

### 7.1. Conclusiones

En el presente proyecto se ha diseñado, implementado y evaluado un algoritmo de aprendizaje por refuerzo para sistemas multiagente en los que la cooperación entre agentes es posible mediante comunicación indirecta en un entorno discreto. Se ha diseñado siguiendo la nueva metodología de desarrollo Gaia, extendiéndola allí donde era necesario para la comprensión del sistema. Se ha implementado en la plataforma de agente JADE, en la cual se ha incluido una ontología en la que se basan protocolos de comunicación entre los agentes del sistema.

El sistema multiagente implementado está formado por un total de **12 agentes** y **29 comportamientos** que implementan **17 roles**, estando 5 de esos roles están ligados directamente a la arquitectura. Los agentes usan para comunicarse un total de **26 protocolos** y **37 mensajes**, sin contar los mensajes sin contenido como parte de las confirmaciones. Dichos mensajes pueden contener un total de **18 acciones** y **16 predicados** que a su vez usan **12 conceptos** diferentes. Todo ello está implementado en las **16481 líneas de código** que componen las **134 clases** y **18 interfaces** utilizadas.

JADE, por su parte, siendo una excelente plataforma de agentes, también ha dificultado en algunas partes el desarrollo del proyecto. En particular, la forma de usar los **servicios del agente DF** es mediante una llamada (`DFService.search()`, `DFService.register()`...) que a su vez crea una comunicación asíncrona y bloqueante con dicho agente. El sistema simplifica en gran medida el uso de los servicios de dicho agente, pero cuando el agente no devuelve el mensaje con la información requerida, nuestro agente se queda bloqueado. Esta situación se da apenas en una llamada de cada millón, pero ha de ser tenida en cuenta.

La comunicación y, en especial, **la sincronización entre los agentes** ha sido una de las tareas más difíciles en el presente proyecto. En las pruebas realizadas hasta 16 agentes competían por acceder a una serie de recursos compartidos, siendo moderados por un único agente y teniendo en cuenta que en cualquier momento se puede perder un mensaje o una confirmación. Hacer que dicha sincronización sea ágil para que la resolución del problema ha requerido una carga de trabajo importante.

En este tipo de agentes es importante tener en cuenta **la escalabilidad del sistema**. En la configuración de los experimentos actuales nunca ha habido más de 60 agentes de los cuales sólo 18 solicitaban información de control. Los dos agentes de control aguantan perfectamente la carga de trabajo que se les da en estos experimentos. Sin embargo, sería muy interesante poder ampliar el número de agentes interactuando en el sistema para probar este y otros muchos algoritmos.

Ello implica tener un importante **cuello de botella** en cuanto al control del acceso a los recursos, puesto que un número muy superior de agentes pueden estar intentando interactuar con un único control. Es muy interesante por tanto **distribuir el control** de acceso a los recursos de forma que cada agente forme parte de dicho control. De esta forma se puede evitar sobrecargas debidas al control cuando el número de agentes escala se ve incrementado.

Además, el algoritmo implementado ofrece mejores resultados que el algoritmo de aprendizaje por refuerzo monoagente Q-Learning aplicado a sistemas multiagente, aplicando **técnicas para aumentar el rendimiento** de Q-Learning ( $\epsilon$  – *greedy* variable, retardo del proceso de aprendizaje) y **técnicas que aprovechan la naturaleza multiagente cooperativa** del sistema (conocimiento previo del entorno, conocimiento compartido entre los agentes).

Uno de los resultados más notables es el equilibrio logrado entre las diferentes técnicas utilizadas. Cada una de ellas mejora una parte del algoritmo:

- **Conocimiento previo del entorno:** Gracias a él, los agentes saben por qué camino es más rápido ir, lo que hace que no sea necesaria demasiada exploración. Esta técnica tiene mayor rendimiento cuantos más agentes se encuentran en el entorno en comparación al caso base, puesto que en los entornos con pocos agentes, la tarea de exploración es mucho más fácil de realizar.
- **Retardo del proceso de aprendizaje:** Permite a los agentes obtener un conocimiento del entorno que aún no está demasiado saturado para poder actuar correctamente cuando lo esté. Esta técnica no tiene apenas impacto cuando es la única que se usa, puesto que facilita el aprendizaje de los primeros agentes. Los últimos en entrar al entorno se verán interrumpidos por otros que ya conocen la solución y no la pueden compartir.
- **$\epsilon$  – *greedy* variable:** Evita que la exploración del entorno continúe cuando ya no es necesaria, explotando el conocimiento adquirido. El uso de esta técnica es fundamental para alcanzar el equilibrio entre exploración del entorno y explotación del conocimiento. Sin embargo, cuando es la única que se usa, su rendimiento es bastante discreto.
- **Conocimiento compartido:** Acelera el proceso de aprendizaje, puesto que unos agentes pueden aprender de las experiencias de otros. Su uso es fundamental cuando el número de agentes se incrementa puesto que, en algunos casos,

consigue acercarse al rendimiento obtenido con la mitad de agentes. Sin embargo, si los parámetros por los que se regula esta técnica no son bien configurados, corre el riesgo de ser contraproducente, puesto que todos los agentes pueden llegar a leer y ejecutar una política de acciones errónea.

Pero las técnicas adquieren mucho más valor cuando se conjugan:

- **Retardo del proceso de aprendizaje y conocimiento compartido:** Unos pocos agentes empiezan a explorar el entorno, aprendiendo cómo reaccionar ante situaciones en las que otros agentes les interrumpen el paso en puntos clave del entorno. Siendo sólo unos pocos la solución es fácil, cosa que no pasaría si estuvieran todos presentes. La obtención de dicha experiencia sólo tiene sentido si se puede compartir, puesto que todos los agentes aprenderán a evitar dichas interrupciones cuando sólo unos pocos las han experimentado.
- **$\epsilon$  – greedy variable y conocimiento previo del entorno:** Teniendo un conocimiento previo del entorno, no es necesaria tanta exploración del entorno, por lo que los agentes pueden concentrarse en aprender el resultado de sus interacciones con otros agentes. Es por ello necesario regular el equilibrio entre exploración y explotación de forma dinámica.
- **Conocimiento compartido y conocimiento previo del entorno:** De nuevo, teniendo un conocimiento previo del entorno según el que saben cómo llegar a la meta, los agentes pueden concentrarse en compartir la información adquirida acerca de las interrupciones entre ellos y en aprender a cooperar para una obtención más rápida de la solución.

Aplicando la política de decisión de la acción a tomar  $\epsilon$  – greedy variable en principio es de esperar que lo ideal sea que todos los agentes alcancen una política totalmente avariciosa, en la que sólo ejecutarán acciones conocidas. Pero los resultados indican que  $\epsilon$  toma valores cercanos a 0.05 (5% de acciones aleatorias, 95% de acciones conocidas), el sistema sufre un decrecimiento súbito del rendimiento, llegando muy pocos agentes a completar su tarea en cada episodio. Es por eso que una de las conclusiones del presente trabajo es que las **políticas de decisión de la acción deterministas no funcionan correctamente aplicadas a este tipo de sistemas multiagente**. Es necesario mantener siempre un mínimo de aleatoriedad en la decisión de la acción.

## 7.2. Trabajos futuros

En el presente trabajo no se ha tratado en ningún momento el tema de la **aproximación de funciones** para el cálculo de las tablas Q de los diferentes algoritmos. Es un tema muy interesante, puesto que muchas veces la acción a tomar sólo depende de uno o dos datos del estado, siendo el resto irrelevantes. Un aproximador de funciones como una red

neuronal es capaz de discernir estas circunstancias, de forma que acelerarían el proceso de aprendizaje.

Desde el principio se ha tratado con un entorno discreto con el objetivo de simplificar la tarea de aprendizaje. Sin embargo, la posibilidad de aplicar este tipo de algoritmos a un **entorno continuo** los acerca a su aplicación en la vida real.

Una línea de investigación importante es el uso de la política de decisión del movimiento  $\epsilon$  – *greedy variable* con parámetros y funciones no lineales. El hecho de que la evolución de las probabilidades sea lineal con respecto al número de episodios ha podido limitar el rendimiento de esta técnica. Además, sería interesante estudiar algún tipo de evolución dependiente del proceso de aprendizaje de todos los agentes y no sólo del que ejecuta la política.

El uso del **conocimiento compartido** es una parte fundamental del trabajo. Su aplicación se ha regido por valores que miden si los objetivos del proceso de aprendizaje de un agente se están cumpliendo. En caso de que no se estén cumpliendo, se elige la acción con mayor valoración por el resto de agentes. Dicha elección puede ser realizada por otros criterios como, por ejemplo, elegir la acción más veces recomendada por los agentes sin tener en cuenta su valoración. Además, los agentes pueden llegar a implementar algoritmos que impliquen la comunicación directa entre ellos, liberando de dicha carga al sistema.

Sería interesante, además, probar su aplicación a sistemas en los que intervengan un **número mayor de agentes** y poder así comparar los resultados con los obtenidos por otros algoritmos de aprendizaje por refuerzo multiagente. El tiempo que toma el algoritmo crece con el número de agentes presentes en el sistema, llegando a las 20 horas en alguna de las pruebas realizadas usando sólo una de las técnicas planteadas con 16 agentes en el sistema.

En ningún momento se ha hecho ninguna consideración acerca del sistema como **sistema distribuido**. Sin embargo, la plataforma JADE ofrece grandes posibilidades en este sentido, que deberán ser explotadas para la aplicación de este tipo de algoritmos a problemas de la vida real.

En cuanto al control de la aplicación, en la actualidad se basa en un **control centralizado**. Existen dos agentes encargados de dar paso a los agentes en cada proceso de aprendizaje. Concretamente, en el control del proceso de aprendizaje en entorno compartido, un total de 16 agentes están constantemente interactuando con un único control.

En caso de extender el trabajo a muchos más agentes, la existencia del control centralizado del proceso de aprendizaje supone un **cuello de botella** importante en el sistema tanto en tiempo como en memoria. En **tiempo de ejecución**, puesto que el control centralizado

estará recibiendo peticiones constantemente que tendrá que procesar, retrasando el inicio del agente que realmente tiene que entrar a ejecutar. Y en **memoria**, puesto que las estructuras de control que tiene que mantener un solo agente serán mucho más complejas cuando el número de agentes crezca.

Por ello, sería muy interesante adoptar un **control descentralizado** en el que la información de control estuviera distribuida entre todos los agentes. De esta forma todos los agentes saben en todo momento cuando han de ejecutar y sólo tienen que esperar pacientemente su turno. Además, las estructuras de control también pueden estar distribuidas, de forma que su crecimiento sería soportado por un número igualmente creciente de agentes.





## 8. Referencias

- [1] ANA MAS, *Agentes software y sistemas multiagente. Conceptos, arquitecturas y aplicaciones*. Pearson/Prentice Hall, 2004.
- [2] Tom M. Mitchell, *Machine Learning*. WCB/McGraw-Hill, 1997.
- [3] Daniel Borrajo Millán, Jesús González Boticario y Predo Isasi Viñuela. *Aprendizaje Automático*. Editorial Sanz y Torres, 2006.
- [4] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns*, Addison-Wesley: Reading, MA, 1995.
- [5] Gerhard Weiss, *Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 1999.
- [6] Wooldridge, M., Jennings, N.R. y Kinny D. 2000. *The Gaia methodology for agent-oriented analysis and design*. J. Autonom. Agents Multi-Agent Syst. 3, 3, 285–312.
- [7] Zambonelli, F., Jennings, N. and Wooldridge, M., 2003, *Developing multiagent systems: the Gaia methodology*, ACM Transaction on Software Engineering and Methodology, 12(3), 317-370
- [8] Collis, J. and Ndumu, D.: *Zeus Methodology Documentation Part I: The Role Modelling Guide*. Intelligent Systems Research Group, BT labs. British Telecommunications (1999)
- [9] C. Iglesias, M. Garijo, J. C. Gonz\_alez, and J. R. Velasco. *Analysis and design of multiagent systems using MAS-CommonKADS*,. in Intelligent Agents IV (LNAI Volume 1365), M. P. Singh, A. Rao, and M. J. Wooldridge, (Eds.), Springer-Verlag: Berlin, Germany, 1998, pp. 313.326.
- [10] S. DeLoach, M. Wood and C. Sparkman, *Multiagent Systems Engineering*, The International Journal of Software Engineering and Knowledge Engineering, Volume 11 no. 3, June 2001
- [11] DeLoach, S. 2001. *Analysis and Design using MaSE and agentTool*. Proceedings of the 12th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS).
- [12] Carlos A. Iglesias, Mercedes Garijo, José C. González y Juan R. Velasco, 1998. *Analysis and Design of Multiagent Systems using MAS-CommonKADS*.
- [13] Fabio Bellifemine, Giovani Caire y Dominic Greenwood. *Developing multi-agent systems with JADE*. John Wiley & Sons, Ltd, 2007.
- [14] Bellifemine, F., Caire, G., Trucco, T., Rimassa, G.: *Jade Programmer's Guide*. JADE 2.5 <http://sharon.cselt.it/projects/jade/>, 2002
- [15] Nikolaos Spanoudakis y Pavlos Moraitis. *Gaia Agents Implementation through Models Transformation*. Paris Descartes University, 2002.
- [16] [Moraitis et al., 2003a] Moraitis, P., Petraki, E., Spanoudakis, N.: *Engineering JADE Agents with the Gaia Methodology*. In R. Kowalszyk, et al. (eds), "Agent Technologies, Infrastructures, Tools, and Applications for e-Services", LNAI 2592, Springer-Verlag, 2003, pp. 77-91
- [17] [FIPA, 2000] *FIPA specifications*. <http://www.fipa.org/repository/standardspecs.html>.
- [18] *Jade tutorial application-defined content languages and ontologies usage*. Giovanni Caire y David Cabanillas 2008. Telecom Italia.

- [19] *Reinforcement Learning: A Survey*. Leslie Pack Kaelbling, Michael L. Littman y Andrew W. Moore. Journal of Artificial Intelligence Research 4 (1996) 237-285.
- [20] *Reinforcement Learning: An Introduction*. Richard S. Sutton y Andrew G. Barto. The MIT Press, 2005,
- [21] Fernando Fernández y Daniel Borrajo. *Aprendizaje Automático: Conceptos básicos y avanzados. Capítulo 11: Aprendizaje Automático*. Pearson Prentice Hall, 2006.
- [22] *A Comprehensive Survey of Multi-Agent Reinforcement Learning*. Lucian Busoniu, Robert Babuska y Bart De Schutter. Delft University of Technology, 2005. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews.
- [23] Panait, L. and Luke, S. (2005). *Cooperative multi-agent learning: The state of the art*. Autonomous Agents and Multi-Agent Systems, 11(3):387-434.
- [24] S. Sen and G. Weiss. *Learning in multiagent systems*, in Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, G. Weiss, Ed. MIT Press, 1999, ch. 6, pp. 259.298.
- [25] J. Hu and M. P. Wellman. *Multiagent reinforcement learning: Theoretical framework and an algorithm*,. in Proceedings Fifteenth International Conference on Machine Learning (ICML-98), Madison, US, 24.27 July 1998, pp. 242.250.
- [26] M. Lauer and M. Riedmiller. *An algorithm for distributed reinforcement learning in cooperative multi-agent systems*, in Proceedings Seventeenth International Conference on Machine Learning (ICML-00), Stanford University, US, 29 June 2 July 2000, pp. 535.542.
- [27] V. Conitzer and T. Sandholm. *AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents*,. in Proceedings Twentieth International Conference on Machine Learning (ICML-03), Washington, US, 21.24 August 2003, pp. 83.90.
- [28] Metodología de desarrollo INGENIAS. <http://grasia.fdi.ucm.es/main/node/67>
- [29] C. Guestrin, M. G. Lagoudakis, and R. Parr. *Coordinated reinforcement learning*,. in Proceedings Nineteenth International Conference on Machine Learning (ICML-02), Sydney, Australia, 8.12 July 2002, pp. 227.234.
- [30] M. L. Littman, "Value-function reinforcement learning in Markov games," Journal of Cognitive Systems Research, vol. 2, no. 1, pp. 55– 66, 2001.
- [31] JADEX, <http://vsis-www.informatik.uni-hamburg.de/projects/jadex/features.php>
- [32] JASON, <http://jason.sourceforge.net/Jason/Jason.html>

# ANEXO I: MODELO DE ROLES

## I.1. Modelo preeliminar de roles

### I.1.1. Gestión del entorno Tablero

Gestor de Estado	
<b>Descripción</b>	Lee y extrae la información que describe el estado en el que se sitúa un agente.
<b>Protocolos y actividades</b>	PedirEstado
<b>Permisos</b>	Puede leer las posiciones de los jugadores en el entorno Entorno. Puede leer las posiciones de los obstáculos en el entorno Entorno.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
GESTOR_DE_ESTADO = ([PedirEstado]) <sup>w</sup>	
<b>Seguridad</b>	Establece conexiones con el entorno Tablero.

Tabla 18 : Rol preliminar Gestor de Estado

Gestor de reinicio	
<b>Descripción</b>	Reinicia la posición de un agente cuando este lo requiere,
<b>Protocolos y actividades</b>	PedirReinicio
<b>Permisos</b>	Puede leer, escribir y borrar las posiciones de los jugadores en el entorno Entorno. Puede leer las posiciones de los inicios en el entorno Entorno.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
GESTOR_DE_REINICIO = ([PedirReinicio]) <sup>w</sup>	
<b>Seguridad</b>	Establece conexiones con el entorno Tablero.

Tabla 19 : Rol preliminar Gestor de Reinicio

Gestor de Movimiento	
<b>Descripción</b>	Ejecuta los movimientos sobre Entorno que los jugadores necesitan para su proceso de aprendizaje y les devuelve el refuerzo relativo al estado que ocupan.
<b>Protocolos y actividades</b>	PedirMovimiento.
<b>Permisos</b>	Puede leer, escribir y borrar las posiciones de los jugadores en el entorno Entorno. Puede leer las posiciones de las metas y de los obstáculos en el entorno Entorno.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
GESTOR_DE_MOVIMIENTO = ([PedirMovimiento]) <sup>w</sup>	
<b>Seguridad</b>	Establece conexiones con el entorno Tablero.

Tabla 20 : Rol preliminar Gestor de Movimiento

### 1.1.2. Gestión del entorno Estadístico

Gestor Estadístico	
<b>Descripción</b>	Reúne las estadísticas relativas a los procesos de aprendizaje de los diferentes agentes.
<b>Protocolos y actividades</b>	EnviarEstadísticas.
<b>Permisos</b>	Puede escribir en el entorno Estadístico.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
GESTOR_ESTADISTICO = ([EnviarEstadísticas]) <sup>w</sup>	
<b>Seguridad</b>	Establece conexiones con el entorno Estadístico.

Tabla 21 : Rol preliminar Gestor Estadístico

Procesador estadístico	
<b>Descripción</b>	Procesa la estadísticas almacenadas para ser mostradas por pantalla
<b>Protocolos y actividades</b>	ProcesarEstadísticas.
<b>Permisos</b>	Puede leer y escribir en el entorno Estadístico.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
PROCESADOR_ESTADISTICO = ([ProcesarEstadísticas]) <sup>w</sup>	
<b>Seguridad</b>	Establece conexiones con el entorno Estadístico.

Tabla 22 : Rol preliminar Procesador Estadístico

### 1.1.3. Gestión del entorno Repositorio

Gestor de Conocimiento en Entorno Exclusivo	
<b>Descripción</b>	Reúne el conocimiento adquirido por los diferentes procesos de aprendizaje que cuentan con el uso del entorno en exclusiva y lo envía a los agentes que lo soliciten.
<b>Protocolos y actividades</b>	EnviarPolíticaExclusiva, PedirMovimientoExclusivo
<b>Permisos</b>	Puede leer y escribir en la política exclusiva del entorno Repositorio.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
GESTOR_DE_CONOCIMIENTO_EXCLUSIVO = ([ EnviarPolíticaExclusiva]    [PedirMovimientoIndividual]) <sup>w</sup> .	
<b>Seguridad</b>	Establece conexiones con el entorno Repositorio.

Tabla 23 : Rol preliminar Gestor de Conocimiento en Entorno Exclusivo

<b>Gestor de Conocimiento en Entorno Compartido</b>	
<b>Descripción</b>	Reúne el conocimiento adquirido por los diferentes procesos de aprendizaje que cuentan con el uso del entorno compartido con otros agentes y lo envía a los agentes que lo soliciten.
<b>Protocolos y actividades</b>	EnviarPolíticaCompartida, PedirPolíticaCompartida.
<b>Permisos</b>	Puede leer y escribir en la política compartida del entorno Repositorio.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
GESTOR_DE_CONOCIMIENTO_EXCLUSIVO = ([ EnviarPolíticaExclusiva]    [PedirMovimientoIndividual]) <sup>w</sup> .	
<b>Seguridad</b>	Establece conexiones con el entorno Repositorio.

Tabla 24 : Rol preliminar Gestor de Conocimiento en Entorno Compartido

<b>Procesador de Conocimiento</b>	
<b>Descripción</b>	Procesa el conocimiento adquirido por los diferentes procesos de aprendizaje.
<b>Protocolos y actividades</b>	<u>ProcesarConocimiento</u>
<b>Permisos</b>	Puede leer y escribir en la política exclusiva del entorno Repositorio. Puede leer y escribir en la política compartida del entorno Repositorio.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
PROCESADOR_DE_CONOCIMIENTO = ([ProcesarConocimiento]) <sup>w</sup>	
<b>Seguridad</b>	Establece conexiones con el entorno Repositorio.

Tabla 25 : Rol preliminar Procesador de Conocimiento

#### 1.1.4. Jugadores en Entorno Exclusivo

<b>Jugador en Entorno Exclusivo</b>	
<b>Descripción</b>	Adquiere el conocimiento necesario para alcanzar la meta desde cualquier estado teniendo un uso exclusivo del entorno.
<b>Protocolos y actividades</b>	<u>PedirRegistroEnEntorno</u> , <u>EntrenamientoExclusivo</u> , <u>EntrenamientoExclusivoAcabado</u> , <u>EnviarConocimientoExclusivo</u> .
<b>Permisos</b>	Puede establecer conexiones con los roles gestor de estado, gestor de movimiento, gestor de reinicio, control de aprendizaje en entorno exclusivo, gestor estadístico y gestor de conocimiento exclusivo.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
JUGADOR_INDIVIDUAL = <u>PedirRegistroEnEntorno</u> , <u>EntrenamientoExclusivo</u> , <u>EntrenamientoExclusivoAcabado</u> , <u>EnviarConocimientoExclusivo</u>	
<b>Seguridad</b>	Conexiones establecidas con los roles gestor de estado, gestor de movimiento, gestor de reinicio, control de aprendizaje en entorno exclusivo, gestor estadístico y gestor de conocimiento exclusivo.

Tabla 26 : Rol preliminar Jugador en Entorno Exclusivo

### 1.1.5. Jugadores en Entorno Compartido

Jugador en Entorno Compartido	
<b>Descripción</b>	Adquiere el conocimiento necesario para alcanzar la meta desde cualquier estado en un entorno compartido con otros agentes con el mismo objetivo
<b>Protocolos y actividades</b>	PedirRegistroEnEntorno, EntrenamientoCompartido, AcabadoCicloEntrenamientoCompartido.
<b>Permisos</b>	Puede establecer conexiones con los roles gestor de estado, gestor de movimiento, gestor de reinicio, control de aprendizaje en entorno compartido, gestor estadístico, gestor de conocimiento exclusivo y gestor de conocimiento compartido.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
JUGADOR_EN_ENTORNO_COMPARTIDO = PedirRegistroEnEntorno. EntrenamientoCompartido. AcabadoCicloEntrenamientoCompartido	
<b>Seguridad</b>	Conexiones establecidas con los roles gestor de estado, gestor de movimiento, gestor de reinicio, control de aprendizaje en entorno compartido, gestor estadístico, gestor de conocimiento exclusivo y gestor de conocimiento compartido.

Tabla 27 : Rol preliminar Jugador en Entorno Compartido

### 1.1.6. Control de los procesos de aprendizaje

Control del Entrenamiento en Entorno Exclusivo	
<b>Descripción</b>	Reúne la información acerca del estado de todos los procesos de aprendizaje en entorno exclusivo existentes y gestiona ordenadamente el acceso a los recursos necesario para dichos procesos.
<b>Protocolos y actividades</b>	PedirRegistroEnEntorno, ListoParaEntrenamientoExclusivo, EntrenamientoExclusivoAcabado, EmpezarEntrenamientoExclusivo,
<b>Permisos</b>	Puede establecer conexiones con los roles Jugador Entorno Exclusivo y Control del Entrenamiento en Entorno Compartido y con el entorno Tablero.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
CONTROL_EXCLUSIVO = ([PedirRegistroEnEntorno]    [ListoParaEntrenamientoExclusivo]    [EntrenamientoExclusivoAcabado]    [EmpezarEntrenamientoExclusivo] ) <sup>w</sup>	
<b>Seguridad</b>	Conexiones establecidas con los roles Jugador Entorno Exclusivo y Control del Entrenamiento en Entorno Compartido y con el entorno Tablero.

Tabla 28 : Rol preliminar Control del Entrenamiento en Entorno Exclusivo

Control del Entrenamiento en Entorno Compartido	
<b>Descripción</b>	Reúne la información acerca del estado de todos los procesos de aprendizaje en entorno compartido existentes y gestiona ordenadamente el acceso a los recursos necesario para dichos procesos.
<b>Protocolos y actividades</b>	PedirRegistroEnEntorno, ListoParaEpisodioEntrenamientoCompartido, AcabadoEpisodioEntrenamientoCompartido, ListoParaCicloEntrenamientoCompartido, AcabadoCicloEntrenamientoCompartido, EntrenamientoCompartidoAcabado, EmpezarEpisodioEntrenamientoCompartido, EmpezarCicloEntrenamientoCompartido, ActualizarGrafica.
<b>Permisos</b>	Puede establecer conexiones con los roles Jugador Entorno Compartido y Gestor Estadístico y con el entorno Tablero.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
CONTROL = ([ PedirRegistroEnEntorno]    [ListoParaEpisodioEntrenamientoCompartido]    [AcabadoEpisodioEntrenamientoCompartido]    [ListoParaCicloEntrenamientoCompartido]    [AcabadoCicloEntrenamientoCompartido]    [EntrenamientoCompartidoAcabado]    [EmpezarEpisodioEntrenamientoCompartido]    [EmpezarCicloEntrenamientoCompartido]    [ActualizarGrafica] ) <sup>w</sup>	
<b>Seguridad</b>	Conexiones establecidas con los roles Jugador Entorno Compartido y con el entorno Tablero

Tabla 29 : Rol preliminar Control del Entrenamiento en Entorno Compartido

## 1.2. Modelo completo de roles

### 1.2.1. Gestión del entorno Tablero

Gestor de Estado	
<b>Descripción</b>	Lee y extrae la información que describe el estado en el que se sitúa un agente.
<b>Protocolos y actividades</b>	PedirEstado, RecibirPeticiónEstado, <u>EncontrarEstado</u> , ResponderEstado, ResponderEstadoDesconocido.
<b>Permisos</b>	Puede leer, escribir y borrar las posiciones de los jugadores en el entorno Entorno. Puede leer las posiciones de los inicios en el entorno Entorno.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
GESTOR_DE_ESTADO = ( [PedirEstado] ) <sup>w</sup> .  PEDIR_ESTADO = RecibirPeticiónEstado. <u>BuscarTipoJugador</u> . [ TratarJugadorExclusivo    TratarJugadorCompartido ]  TRATAR_JUGADOR_EXCLUSIVO = <u>EncontrarEstadoExclusivo</u> . ( [ResponderEstadoExclusivo]    [ResponderEstadoExclusivoDesconocido] )  TRATAR_JUGADOR_COMPARTIDO = <u>EncontrarEstadoCompartido</u> . ( [ResponderEstadoCompartido]    [ResponderEstadoCompartidoDesconocido] )	
<b>Seguridad</b>	Establece conexiones con el entorno Tablero.

Tabla 30 : Rol Gestor de Estado

<b>Gestor de Reinicio</b>	
<b>Descripción</b>	Reinicia la posición de un agente cuando este lo requiere
<b>Protocolos y actividades</b>	PedirReinicio, RecibirPeticiónReinicio, <u>EncontrarInicioLibre</u> , <u>ReiniciarPosiciónAgente</u> , <u>ConfirmarReinicio</u> , ReinicioIncorrecto.
<b>Permisos</b>	Puede leer, escribir y borrar las posiciones de los jugadores en el entorno Entorno. Puede leer las posiciones de los inicios en el entorno Entorno.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
GESTOR_DE_REINICIO = ( [PedirReinicio] ) <sup>ω</sup>	
PEDIR_REINICIO = RecibirPeticiónReinicio. <u>EncontrarInicioLibre</u> . ( (ReiniciarPosiciónAgente . ConfirmarReinicio )    ReinicioIncorrecto )	
<b>Seguridad</b>	Establece conexiones con el entorno Tablero.

Tabla 31 : Rol Gestor de Reinicio

<b>Gestor de Movimiento</b>	
<b>Descripción</b>	Ejecuta los movimientos sobre Entorno que los jugadores necesitan para su proceso de aprendizaje y les devuelve el refuerzo relativo al estado que ocupan.
<b>Protocolos y actividades</b>	PedirMovimiento, RecibirMovimiento, <u>EncontrarEstado</u> , <u>CalcularSiguienteEstado</u> , <u>MoverJugador</u> , MovimientoCorrecto, MovimientoIncorrecto.
<b>Permisos</b>	Puede leer, escribir y borrar las posiciones de los jugadores en el entorno Entorno. Puede leer las posiciones de las metas y de los obstáculos en el entorno Entorno.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
GESTOR_DE_MOVIMIENTO = ( [PedirMovimiento] ) <sup>ω</sup>	
PEDIR_MOVIMIENTO = RecibirMovimiento . <u>EncontrarEstado</u> . <u>CalcularSiguienteEstado</u> . ( (MoverJugador . MovimientoCorrecto )    MovimientoIncorrecto )	
<b>Seguridad</b>	Establece conexiones con el entorno Tablero.

Tabla 32 : Rol Gestor de Movimiento



### 1.2.2. Gestión del entorno Estadístico

Gestor Estadístico	
<b>Descripción</b>	Reúne las estadísticas relativas a los procesos de aprendizaje de los diferentes agentes.
<b>Protocolos y actividades</b>	<u>EnviarEstadísticasExclusivas</u> , <u>RecibirEstadísticasExclusivas</u> , <u>InsertarEstadísticasExclusivas</u> , <u>ResponderEstadísticasExclusivas</u> , <u>EnviarEstadísticasCompartidas</u> , <u>RecibirEstadísticasCompartidas</u> , <u>InsertarEstadísticasCompartidas</u> , <u>ResponderEstadísticasCompartidas</u> .
<b>Permisos</b>	Puede escribir en el entorno Estadístico.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
<p>GESTOR_ESTADISTICO = ( [<u>EnviarEstadísticasExclusivas</u>]    [<u>EnviarEstadísticasCompartidas</u>] )<sup>ω</sup></p> <p>ENVIAR_ESTADISTICAS_EXCLUSIVAS = <u>RecibirEstadísticasExclusivas</u>. <u>InsertarEstadísticasExclusivas</u>. <u>ResponderEstadísticasExclusivas</u></p> <p>ENVIAR_ESTADISTICAS_COMPARTIDAS = <u>RecibirEstadísticasCompartidas</u>. <u>InsertarEstadísticasCompartidas</u>. <u>ResponderEstadísticasCompartidas</u></p>	
<b>Seguridad</b>	Establece conexiones con el entorno Estadístico.

Tabla 33 : Rol Gestor Estadístico

Procesador Estadístico	
<b>Descripción</b>	Procesa la estadísticas almacenadas para ser mostradas por pantalla
<b>Protocolos y actividades</b>	<u>ProcesarEstadísticas</u> , <u>ActualizarGráfica</u> , <u>RecibirPeticiónActualizarGráfica</u> , <u>EjecutarActualizarGráfica</u> , <u>ConfirmarActualizarGráfica</u> .
<b>Permisos</b>	Puede leer y escribir en el entorno Estadístico.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
<p>PROCESADOR_ESTADISTICO = (   [<u>ActualizarGráfica</u>] )<sup>ω</sup></p> <p>ACTUALIZAR_GRAFICA = <u>RecibirPeticiónActualizarGráfica</u> . <u>EjecutarActualizarGráfica</u> . <u>ConfirmarActualizarGráfica</u></p>	
<b>Seguridad</b>	Establece conexiones con el entorno Estadístico.

Tabla 34 : Rol Procesador Estadístico

### 1.2.3. Gestión del entorno Repositorio

Gestor de Conocimiento en Entorno Exclusivo	
<b>Descripción</b>	Reúne el conocimiento adquirido por los diferentes procesos de aprendizaje que cuentan con el uso del entorno en exclusiva y lo envía a los agentes que lo soliciten.
<b>Protocolos y actividades</b>	EnviarPolíticaExclusiva, RecibirPolíticaExclusiva, InsertarPolíticaExclusiva, ConfirmarPolíticaExclusiva, PedirMovimientoExclusivo, RecibirMovimientoExclusivo, BuscarMovimientoExclusivo, ResponderMovimientoExclusivo, MovimientoExclusivoDesconocido.
<b>Permisos</b>	Puede leer y escribir en la política individual del entorno Repositorio.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
<p>GESTOR_DE_CONOCIMIENTO_ENTORNO_EXCLUSIVO = ( [EnviarPolíticaExclusiva]    [PedirMovimientoExclusivo] )<sup>w</sup></p> <p>ENVIAR_POLITICA_EXCLUSIVA = RecibirPolíticaExclusiva. <u>InsertarPolíticaExclusiva</u>. ConfirmarPolíticaExclusiva</p> <p>PEDIR_MOVIMIENTO_EXCLUSIVO = RecibirMovimientoExclusivo. <u>BuscarMovimientoExclusivo</u>. ([ResponderMovimientoExclusivo]    [MovimientoExclusivoDesconocido])</p>	
<b>Seguridad</b>	Establece conexiones con el entorno Repositorio.

Tabla 35 : Rol Gestor de Conocimiento en Entorno Exclusivo

Gestor de Conocimiento en Entorno Compartido	
<b>Descripción</b>	Reúne el conocimiento adquirido por los diferentes procesos de aprendizaje que cuentan con el uso del entorno compartido con otros agentes y lo envía a los agentes que lo soliciten.
<b>Protocolos y actividades</b>	EnviarPolíticaCompartida, RecibirPolíticaCompartida, <u>InsertarPolíticaCompartida</u> , ConfirmarPolíticaCompartida , PedirPolíticaCompartida, RecibirPeticiónPolíticaCompartida, <u>BuscarPolíticaCompartida</u> , ResponderPolíticaCompartida
<b>Permisos</b>	Puede leer y escribir en la política compartida del entorno Repositorio.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
<p>GESTOR_DE_CONOCIMIENTO_ENTORNO_COMPARTIDO = ( [EnviarPolíticaCompartida]    [PedirPolíticaCompartida] )<sup>w</sup></p> <p>ENVIAR_POLITICA_COMPARTIDA = RecibirPolíticaCompartida. <u>InsertarPolíticaCompartida</u>. ConfirmarPolíticaCompartida</p> <p>PEDIR_POLITICA_COMPARTIDA = RecibirPeticiónPolíticaCompartida. <u>BuscarPolíticaCompartida</u>. ResponderPolíticaCompartida</p>	
<b>Seguridad</b>	Establece conexiones con el entorno Repositorio.

Tabla 36 : Rol Gestor de Conocimiento en Entorno Compartido

Procesador de Conocimiento	
<b>Descripción</b>	Procesa el conocimiento adquirido por los diferentes procesos de aprendizaje.
<b>Protocolos y actividades</b>	<u>ProcesarConocimiento</u> , <u>LeerConocimientoCompartido</u> , <u>ProcesarConocimientoCompartido</u> , <u>EscribirConocimientoCompartido</u> .
<b>Permisos</b>	Puede leer y escribir en la política exclusiva del entorno Repositorio. Puede leer y escribir en la política compartida del entorno Repositorio.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
PROCESADOR_DE_CONOCIMIENTO = ([ProcesarConocimiento]) <sup>ω</sup>	
PROCESAR_CONOCIMIENTO = <u>LeerConocimientoCompartido</u> , <u>ProcesarConocimientoCompartido</u> , <u>EscribirConocimientoCompartido</u>	
<b>Seguridad</b>	Establece conexiones con el entorno Repositorio.

Tabla 37 : Rol Procesador de Conocimiento

#### 1.2.4. Jugadores en Entorno Exclusivo

Jugador en Entorno Exclusivo	
<b>Descripción</b>	Adquiere el conocimiento necesario para alcanzar la meta desde cualquier estado teniendo un uso exclusivo del entorno.
<b>Protocolos y actividades</b>	<u>PedirRegistroEnEntorno</u> , <u>EsperarAutorizacion</u> , <u>EntrenamientoExclusivo</u> , <u>ListoParaEntrenamientoExclusivo</u> , <u>EsperarAutorizacionEntrenamientoExclusivo</u> , <u>EpisodioEntrenamientoExclusivo</u> , <u>PedirReinicio</u> , <u>PedirEstado</u> , <u>IniciarEstadísticas</u> , <u>CicloEntrenamientoExclusivo</u> , <u>PedirMovimiento</u> , <u>ModificarMatrizQ</u> , <u>EnviarEstadísticasExclusivas</u> , <u>EntrenamientoExclusivoAcabado</u> , <u>EnviarConocimientoIndividual</u> , <u>FormarPoliticaExclusiva</u> , <u>EnviarPoliticaExclusiva</u> , <u>ConfirmarPoliticaExclusiva</u> .
<b>Permisos</b>	Puede establecer conexiones con los roles gestor de estado, gestor de movimiento, gestor de reinicio, control de aprendizaje en entorno exclusivo, gestor estadístico y gestor de conocimiento exclusivo.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
JUGADOR_ENTORNO_EXCLUSIVO = <u>PedirRegistroEnEntorno</u> . <u>EntrenamientoExclusivo</u> . <u>EnviarConocimientoIndividual</u>	
PEDIR_REGISTRO_ENTORNO = <u>EnviarPeticiónRegistro</u> . <u>EsperarAutorizacion</u>	
ENTRENAMIENTO_EXCLUSIVO = <u>ListoParaEntrenamientoExclusivo</u> . <u>EsperarAutorizacionEntrenamientoExclusivo</u> . ( <u>EpisodioEntrenamientoExclusivo</u> ) <sup>+</sup> . <u>EntrenamientoExclusivoAcabado</u>	
EPISODIO_ENTRENAMIENTO_EXCLUSIVO = <u>PedirReinicio</u> . <u>PedirEstado</u> . <u>IniciarEstadísticas</u> . ( <u>CicloEntrenamientoExclusivo</u> ) <sup>+</sup> . <u>EnviarEstadísticasExclusivas</u>	
CICLO_ENTRENAMIENTO_EXCLUSIVO = <u>PedirMovimiento</u> . <u>PedirEstado</u> . <u>ModificarMatrizQ</u>	
ENVIAR_CONOCIMIENTO_EXCLUSIVO = <u>FormarPoliticaExclusiva</u> . <u>EnviarPoliticaExclusiva</u> . <u>ConfirmarPoliticaExclusiva</u>	
<b>Seguridad</b>	Conexiones establecidas con los roles gestor de estado, gestor de movimiento, gestor de reinicio, control de aprendizaje en entorno exclusivo, gestor estadístico y gestor de conocimiento exclusivo.

Tabla 38 : Rol Jugador en Entorno Exclusivo

### 1.2.5. Jugadores en Entorno Compartido

<b>Jugador en Entorno Compartido</b>	
<b>Descripción</b>	Adquiere el conocimiento necesario para alcanzar la meta desde cualquier estado en un entorno compartido con otros agentes con el mismo objetivo
<b>Protocolos y actividades</b>	PedirRegistroEnEntorno, EntrenamientoCompartido, <u>IniciarProbabilidades</u> , EpisodioEntrenamientoCompartido, ListoParaEpisodioEntrenamientoCompartido, <u>EsperarAutorizacionEpisodioEntrenamientoCompartido</u> , PedirReinicio, PedirEstado, IniciarEstadísticas, CicloEntrenamientoCompartido, ListoParaCicloEntrenamientoCompartido, <u>EsperarAutorizacionCicloEntrenamientoCompartido</u> , PedirMovimiento, <u>ModificarMatrizQ</u> , AcabadoCicloEntrenamientoCompartido, EnviarPoliticaCompartida, EnviarEstadísticas, IntercambiarPoliticaCompartida, AcabadoEpisodioEntrenamientoCompartido.
<b>Permisos</b>	Puede establecer conexiones con los roles gestor de estado, gestor de movimiento, gestor de reinicio, control de aprendizaje en entorno compartido, gestor estadístico, gestor de conocimiento exclusivo y gestor de conocimiento compartido.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
<p>JUGADOR_EN_ENTORNO_COMPARTIDO = PedirRegistroEnEntorno. EntrenamientoCompartido. AcabadoCicloEntrenamientoCompartido</p> <p>ENTRENAMIENTO_COMPARTIDO= <u>IniciarProbabilidades</u> . (EpisodioEntrenamientoCompartido)+</p> <p>EPISODIO_ENTRENAMIENTO_COMPARTIDO = ListoParaEpisodioEntrenamientoCompartido . <u>EsperarAutorizacionEpisodioEntrenamientoCompartido</u> . PedirReinicio . PedirEstado . IniciarEstadísticas . (CicloEntrenamientoCompartido)+ . EnviarPoliticaCompartida. EnviarEstadísticas . [IntercambiarPoliticaCompartida] . AcabadoEpisodioEntrenamientoCompartido</p> <p>CICLO_ENTRENAMIENTO_COMPARTIDO = ListoParaCicloEntrenamientoCompartido . <u>EsperarAutorizacionCicloEntrenamientoCompartido</u> . PedirMovimiento . PedirEstado . <u>ModificarMatrizQ</u> . AcabadoCicloEntrenamientoCompartido</p> <p>ENVIAR_POLITICA_COMPARTIDA =CalcularPolíticaCompartida . EnviarPoliticalCompartida . ConfirmarPoliticaCompartida</p> <p>INTERCAMBIAR_POLITICA_COMPARTIDA = PedirPoliticaCompartida . RecibirPoliticaCompartida</p>	
<b>Seguridad</b>	Conexiones establecidas con los roles gestor de estado, gestor de movimiento, gestor de reinicio, control de aprendizaje en entorno compartido, gestor estadístico, gestor de conocimiento exclusivo y gestor de conocimiento compartido.

Tabla 39 : Rol Jugador en Entorno Compartido

### 1.2.6. Control de los procesos de aprendizaje

Control del Entrenamiento en Entorno Exclusivo	
<b>Descripción</b>	Reúne la información acerca del estado de todos los procesos de aprendizaje en entorno exclusivo existentes y gestiona ordenadamente el acceso a los recursos necesario para dichos procesos.
<b>Protocolos y actividades</b>	PedirRegistroEnEntorno, ListoParaEntrenamientoExclusivo, EntrenamientoExclusivoAcabado, <u>EmpezarEntrenamientoExclusivo</u>
<b>Permisos</b>	Puede establecer conexiones con los roles Jugador Entorno Exclusivo y Control del Entrenamiento en Entorno Compartido y con el entorno Tablero.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
<p>CONTROL_EXCLUSIVO = ( [PedirRegistroEnEntorno]     [ListoParaEntrenamientoExclusivo]    [EntrenamientoExclusivoAcabado]     [<u>EmpezarEntrenamientoExclusivo</u>] )<sup>w</sup></p> <p>PEDIR_REGISTRO_EN_ENTORNO = RecibirPeticiónRegistro. <u>RegistrarJugador</u>. ( [AgenteRegistrado]    [AgenteNoRegistrado] )</p> <p>LISTO_PARA_ENTRENAMIENTO_EXCLUSIVO = RecibirInformeListoExclusivo . <u>NuevoJugadorExclusivoListo</u> . ConfirmarInformeListoExclusivo . <u>ComprobarInicioEntrenamientoExclusivo</u> . [<u>EmpezarEntrenamientoExclusivo</u>]</p> <p>ENTRENAMIENTO_EXCLUSIVO_ACABADO = RecibirInformeAcabadoExclusivo . <u>NuevoJugadorIndividualExclusivo</u> . ConfirmarInformeAcabadoExclusivo . ((<u>ComprobarInicioEntrenamientoExclusivo</u> . [<u>EmpezarEntrenamientoExclusivo</u>] )    (<u>ComprobarInicioEntrenamientoCompartido</u> . [<u>EmpezarEntrenamientoCompartido</u>])</p> <p>EMPEZAR_ENTRENAMIENTO_EXCLUSIVO = <u>InformarEmpezarEntrenamientoExclusivo</u> . <u>ConfirmarInformeEmpezarEntrenamientoExclusivo</u></p>	
<b>Seguridad</b>	Conexiones establecidas con los roles Jugador Entorno Exclusivo y Control del Entrenamiento en Entorno Compartido y con el entorno Tablero.

Tabla 40 : Rol Control del Entrenamiento en Entorno Exclusivo

Control del Entrenamiento en Entorno Compartido	
<b>Descripción</b>	Reúne la información acerca del estado de todos los procesos de aprendizaje colectivo existentes y gestiona ordenadamente el acceso a los recursos necesario para dichos procesos.
<b>Protocolos y actividades</b>	PedirRegistroEnEntorno, ListoParaEpisodioEntrenamientoCompartido, AcabadoEpisodioEntrenamientoCompartido, ListoParaCicloEntrenamientoCompartido, AcabadoCicloEntrenamientoCompartido, EntrenamientoCompartidoAcabado, <u>EmpezarEpisodioEntrenamientoCompartido</u> , <u>EmpezarCicloEntrenamientoCompartido</u> , <u>ActualizarGrafica</u>
<b>Permisos</b>	Puede establecer conexiones con los roles jugador individual, jugador colectivo y procesador estadístico.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
<p>CONTROL_COMPARTIDO = ( [PedirRegistroEnEntorno]     [ListoParaEpisodioEntrenamientoCompartido]     [AcabadoEpisodioEntrenamientoCompartido]     [ListoParaCicloEntrenamientoCompartido]    [AcabadoCicloEntrenamientoCompartido]     [EntrenamientoCompartidoAcabado]    [<u>EmpezarEpisodioEntrenamientoCompartido</u>]     [<u>EmpezarCicloEntrenamientoCompartido</u>]    [ <u>ActualizarGrafica</u>] )<sup>w</sup></p>	

<p>PEDIR_REGISTRO_EN_ENTORNO = RecibirPeticiónRegistro. <u>RegistrarJugador</u>. ( [AgenteRegistrado]    [AgenteNoRegistrado] )</p> <p>LISTO_PARA_EPISODIO_ENTRENAMIENTO_COMPARTIDO =  RecibirInformeListoEpisodioCompartido . <u>NuevoJugadorCompartidoListoEpisodio</u> .  ConfirmarInformeListoEpisodioCompartido .  ComprobarInicioEpisodioEntrenamientoCompartido .  [EmpezarEpisodioEntrenamientoCompartido]</p> <p>ACABADO_EPISODIO_ENTRENAMIENTO_COMPARTIDO =  RecibirInformeAcabadoEpisodioCompartido . <u>NuevoJugadorCompartidoAcabadoEpisodio</u> .  ConfirmarInformeAcabadoEpisodioCompartido .  ComprobarInicioEpisodioEntrenamientoCompartido .  [EmpezarEpisodioEntrenamientoCompartido]</p> <p>LISTO_PARA_CICLO_ENTRENAMIENTO_COMPARTIDO =  RecibirInformeListoCicloCompartido . <u>NuevoJugadorCompartidoListoCiclo</u> .  ConfirmarInformeListoCicloCompartido . <u>ComprobarInicioCicloEntrenamientoCompartido</u> .  [EmpezarCicloEntrenamientoCompartido]</p> <p>ACABADO_CICLO_ENTRENAMIENTO_COMPARTIDO =  RecibirInformeAcabadoCicloCompartido . <u>NuevoJugadorCompartidoAcabadoCiclo</u> .  ConfirmarInformeAcabadoCicloCompartido . <u>ComprobarInicioCicloEntrenamientoCompartido</u> .  [EmpezarCicloEntrenamientoCompartido]</p> <p>ENTRENAMIENTO_COMPARTIDO_ACABADO = RecibirInformeAcabadoCompartido .  <u>NuevoJugadorCompartidoAcabado</u> . ConfirmarInformeAcabadoCompartido .  <u>ComprobarFinalEntrenamientoCompartido</u></p> <p>EMPEZAR_EPISODIO_ENTRENAMIENTO_COMPARTIDO =  InformarEmpezarEpisodioEntrenamientoCompartido .  ConfirmarInformeEmpezarEpisodioEntrenamientoCompartido</p> <p>EMPEZAR_CICLO_ENTRENAMIENTO_COMPARTIDO =  InformarEmpezarCicloEntrenamientoCompartido .  ConfirmarInformeEmpezarCicloEntrenamientoCompartido</p> <p>ACTUALIZAR_GRAFICA = InformarActualizarGrafica . <u>ConfirmarActualizarGrafica</u></p>	
<b>Seguridad</b>	Conexiones establecidas con los roles Jugador Entorno Compartido y con el entorno Tablero

Tabla 41 : Rol Control del Entrenamiento en Entorno Compartido

### 1.2.7. Roles ligados a la arquitectura

Registrador DF	
<b>Descripción</b>	Registra un agente en el agente DF de JADE.
<b>Protocolos y actividades</b>	<u>RegistrarEnDF</u> , <u>EnviarRegistroEnDF</u>
<b>Permisos</b>	Registra agentes en el agente DF de JADE.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
REGISTRADOR_DF = <u>RegistrarEnDF</u>	
REGISTRAR_EN_DF = <u>EnviarRegistroEnDF</u>	
<b>Seguridad</b>	Conexión establecida con el agente DF de JADE

Tabla 42 : Rol Registrador DF

Buscador de Agentes	
<b>Descripción</b>	Busca agentes según su registro en el agente DF de JADE:
<b>Protocolos y actividades</b>	EncontrarAgente, <u>BuscarAgenteEnDF</u>
<b>Permisos</b>	Obtiene registros de agentes del agente DF de JADE.
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
BUSCADOR_DE_AGENTES = <u>EncontrarAgente</u>	
<u>ENCONTRAR_AGENTE</u> = <u>BuscarAgenteEnDF</u>	
<b>Seguridad</b>	Conexión establecida con el agente DF de JADE

Tabla 43 : Rol Buscador de Agentes

Servidor	
<b>Descripción</b>	Sirve algún tipo de informacion.
<b>Protocolos y actividades</b>	SolicitudDeUso, RecibirSolicitudDeUso, <u>AutorizarAgente</u> , AgenteAutorizado, AgenteNoAutorizado
<b>Permisos</b>	Ninguno
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
SERVIDOR = (SolicitudDeUso) <sup>ω</sup>	
SOLICITUD_DE_USO = RecibirSolicitudDeUso . <u>AutorizarAgente</u> . ( [AgenteAutorizado]    [AgenteNoAutorizado] )	
<b>Seguridad</b>	Conexiones establecidas con los agentes que requieran su información

Tabla 44 : Rol Servidor

Procesador	
<b>Descripción</b>	Procesa algún tipo de informacion.
<b>Protocolos y actividades</b>	EsNecesarioProcesar, Procesar
<b>Permisos</b>	Ninguno
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
PROCESADOR = (EsNecesarioProcesar . [Procesar]) <sup>ω</sup>	
<b>Seguridad</b>	Conexiones establecidas con los entornos que tengan información a procesar.

Tabla 45 : Rol Procesador

Finalizable	
<b>Descripción</b>	Espera la orden de finalizar la ejecución y finaliza la ejecución de un agente cuando es necesario.
<b>Protocolos y actividades</b>	FinalizarEjecucion, AgenteFinalizado, AgenteNoFinalizado
<b>Permisos</b>	Ninguno
<b>Responsabilidades</b>	
<b>Ciclo de vida</b>	
PROCESADOR = (FinalizarEjecucion. [AgenteFinalizado    AgenteNoFinalizado]) <sup>ω</sup>	
<b>Seguridad</b>	Conexión establecida con el agente que finaliza la ejecución.

Tabla 46 : Rol Finalizable





## ANEXO II: Modelo de interacciones

### II.1. Modelo preliminar de interacciones

#### II.1.1. Iteraciones con la organización Tablero

Pedir Estado		
Jugador	Tablero	Agente
Un agente Jugador solicita información acerca del estado en el que se encuentra.		Estado jugador

Tabla 47 : Interacción preliminar Pedir Estado

Pedir Reinicio		
Jugador	Tablero	Agente, equipo
Un agente jugador solicita que su estado sea reiniciado a un estado inicial		

Tabla 48 : Interacción preliminar Pedir Reinicio

Pedir Movimiento		
Jugador	Tablero	Agente, movimiento
Un agente jugador solicita un movimiento en el entorno.		

Tabla 49 : Interacción preliminar Pedir Movimiento

#### II.1.2. Interacciones con la organización Estadística

Enviar Estadísticas		
Jugador	Estadístico	Agente, estadísticas
Un agente jugador envía las estadísticas generadas durante un episodio de su proceso de aprendizaje		

Tabla 50 : Interacción preliminar Enviar Estadísticas

#### II.1.3. Interacciones con la organización Repositorio

Enviar Conocimiento		
Jugador	Repositorio	Agente, política, equipo
Un agente jugador, perteneciente a un equipo, envía la política de movimientos generada como consecuencia de su proceso de aprendizaje.		

Tabla 51 : Interacción preliminar Enviar Conocimiento

<b>Pedir Conocimiento Exclusivo</b>		
<b>Jugador Entorno Compartido</b>	<b>Repositorio</b>	Agente, Estado, equipo
Un agente jugador, perteneciente a un equipo, pide conocimiento obtenido a través de jugadores en entorno exclusivo para un estado concreto.		Movimiento exclusivo

Tabla 52 : Interacción preliminar Pedir Conocimiento Exclusivo

<b>Pedir Política Compartida</b>		
<b>Jugador en Entorno Compartido</b>	<b>Repositorio</b>	Agente, equipo
Un agente jugador, perteneciente a un equipo, pide conocer la política obtenida por jugadores en entorno compartido que describe las mejores acciones conocidas.		Política compartida

Tabla 53 : Interacción preliminar Pedir Política Compartida

#### II.1.4. Interacciones con la organización de Jugadores en Entorno Exclusivo

<b>Empezar Entrenamiento Exclusivo</b>		
<b>Control</b>	<b>Jugador en Entorno Exclusivo</b>	Agente
Control entrenamiento en entorno exclusiva informa a un jugador de que puede empezar a ejecutar su proceso de aprendizaje.		

Tabla 54 : Interacción preliminar Empezar Entrenamiento Exclusivo

#### II.1.5. Interacciones con la organización de Jugadores en Entorno Compartido

<b>Empezar Episodio de Entrenamiento Compartido</b>		
<b>Control del entrenamiento en entorno Compartido</b>	<b>Jugador en Entorno Compartido</b>	Agente
Control del entrenamiento en entorno Compartido informa a un jugador en entorno compartido de que puede empezar a ejecutar un episodio de su proceso de aprendizaje.		

Tabla 55 : Interacción preliminar Empezar Episodio de Entrenamiento Compartido

<b>Empezar Ciclo de Entrenamiento Compartido</b>		
<b>Control del entrenamiento en entorno Compartido</b>	<b>Jugador en Entorno Compartido</b>	Agente
Control del entrenamiento en entorno Compartido informa a un jugador en entorno compartido de que puede empezar a ejecutar un ciclo de su proceso de aprendizaje.		

Tabla 56 : Interacción preliminar Empezar Ciclo de Entrenamiento Compartido

### II.1.6. Interacciones con la organización de control

<b>Listo para Entrenamiento Exclusivo</b>		
<b>Jugador en Entorno Exclusivo</b>	<b>Control del entrenamiento en Entorno Exclusivo</b>	Agente
Un jugador en entorno exclusivo informa a Control del entrenamiento en entorno exclusivo de que se encuentra listo para ejecutar su proceso de aprendizaje		

Tabla 57 : Interacción preliminar Listo para Entrenamiento Exclusivo

<b>Entrenamiento Exclusivo Acabado</b>		
<b>Jugador en Entorno Exclusivo</b>	<b>Control del entrenamiento en Entorno Exclusivo</b>	Agente
Un jugador en entorno exclusivo informa a control del entrenamiento en entorno exclusivo de que ha acabado su proceso de aprendizaje.		

Tabla 58 : Interacción preliminar Entrenamiento Exclusivo Acabado

<b>Listo para Episodio de Entrenamiento Compartido</b>		
<b>Jugador en Entorno Compartido</b>	<b>Control del entrenamiento en Entorno Compartido</b>	Agente
Un jugador en entorno compartido informa a Control del entrenamiento en entorno compartido de que se encuentra preparado para ejecutar un episodio de su proceso de aprendizaje.		

Tabla 59 : Interacción preliminar Listo para Episodio de Entrenamiento Compartido

<b>Acabado Episodio de Entrenamiento Compartido</b>		
<b>Jugador en Entorno Compartido</b>	<b>Control del entrenamiento en Entorno Compartido</b>	Agente
Un jugador en entorno compartido informa a Control del entrenamiento en entorno compartido de que ha acabado un episodio de su proceso de aprendizaje.		

Tabla 60 : Interacción preliminar Acabado Episodio de Entrenamiento Compartido

<b>Listo para Ciclo de Entrenamiento Compartido</b>		
<b>Jugador en Entorno Compartido</b>	<b>Control del entrenamiento en Entorno Compartido</b>	Agente
Un jugador en entorno compartido informa a Control del entrenamiento compartido de que se encuentra preparado para ejecutar un ciclo de su proceso de aprendizaje.		

Tabla 61 : Interacción preliminar Listo para Ciclo de Entrenamiento Compartido

<b>Acabado Ciclo de Entrenamiento Colectivo</b>		
<b>Jugador en Entorno Compartido</b>	<b>Control del entrenamiento en Entorno Compartido</b>	Agente
Un jugador en entorno compartido informa a Control del entrenamiento en entorno compartido de que ha acabado un ciclo de su proceso de aprendizaje.		

Tabla 62 : Interacción preliminar Acabado Ciclo de Entrenamiento Colectivo

<b>Entrenamiento Colectivo Acabado</b>		
<b>Jugador en Entorno Compartido</b>	<b>Control del entrenamiento en Entorno Compartido</b>	Agente
Un jugador en entorno compartido informa a Control del entrenamiento en entorno compartido de que ha acabado su proceso de aprendizaje.		

Tabla 63 : Interacción preliminar Entrenamiento Colectivo Acabado

<b>Pedir Registro en Entorno Tablero</b>		
<b>Jugador</b>	<b>Control</b>	Agente
Un agente jugador solicita ser registrado en el entorno Tablero, en el que ejecutará su proceso de aprendizaje.		Registro

Tabla 64 : Interacción preliminar Pedir Registro en Entorno Tablero

## II.2. Modelo completo de interacciones

### II.2.1. Interacciones con la organización Tablero

<b>Pedir Estado</b>		
<b>Jugador en entorno exclusivo o jugador en entorno compartido</b>	<b>Gestor de Estado</b>	Agente
Un agente jugador solicita información acerca del estado en el que se encuentra.		Estado jugador o Estado desconocido

Tabla 65 : Interacción Pedir Estado

<b>Pedir Reinicio</b>		
<b>Jugador en entorno exclusivo o jugador en entorno compartido</b>	<b>Gestor de reinicio</b>	Agente, equipo
Un agente jugador solicita que su estado sea reiniciado a un estado inicial		Confirmar reinicio o reinicio incorrecto

Tabla 66 : Interacción Pedir Reinicio

<b>Pedir Movimiento</b>		
<b>Jugador en entorno exclusivo o jugador en entorno compartido</b>	<b>Gestor de movimiento</b>	Agente, movimiento
Un agente jugador solicita un movimiento en el entorno.		Movimiento correcto o movimiento incorrecto, Refuerzo

Tabla 67 : Interacción Pedir Movimiento

## II.2.2. Interacciones con la organización Estadística

<b>Enviar Estadísticas Entorno Exclusivo</b>		
<b>Jugador en entorno exclusivo</b>	<b>Gestor estadístico</b>	Agente, estadísticas exclusivas
Un agente jugador en entorno exclusivo envía las estadísticas generadas durante un episodio de su proceso de aprendizaje en entorno exclusivo		Confirmación

Tabla 68 : Interacción Enviar Estadísticas Entorno Exclusivo

<b>Enviar Estadísticas Entorno Compartido</b>		
<b>Jugador en entorno compartido</b>	<b>Gestor estadístico</b>	Agente, estadísticas compartidas
Un agente jugador en entorno compartido envía las estadísticas generadas durante un episodio de su proceso de aprendizaje en entorno compartido		Confirmación

Tabla 69 : Interacción Enviar Estadísticas Entorno Compartido

<b>Actualizar Grafica</b>		
<b>Control del entrenamiento en entorno compartido</b>	<b>Procesador estadístico</b>	Agente
Control del entrenamiento en entorno compartido ordena al gestor estadístico que actualice la gráfica que muestra al usuario		Confirmación

Tabla 70 : Interacción Actualizar Grafica

## II.2.3. Interacciones con la organización Repositorio

<b>Enviar Política Exclusiva</b>		
<b>Jugador en entorno exclusivo</b>	<b>Gestor de conocimiento en entorno Exclusivo</b>	Agente, política exclusiva, equipo
Un agente jugador en entorno exclusivo envía la política de movimientos generada como consecuencia de su proceso de aprendizaje.		Confirmación

Tabla 71 : Interacción Enviar Política Exclusiva

<b>Enviar Política Compartida</b>		
<b>Jugador en entorno compartido</b>	<b>Gestor de conocimiento en entorno Compartido</b>	Agente, política compartida, equipo
Un agente jugador en entorno compartido envía la política de movimientos generada como consecuencia de su proceso de aprendizaje.		Confirmación

Tabla 72 : Interacción Enviar Política Compartida

<b>Pedir Mejor Movimiento en Entorno Exclusivo</b>		
<b>Jugador en entorno compartido</b>	<b>Gestor de conocimiento en entorno Exclusivo</b>	Agente, Estado entorno exclusivo, equipo
Un agente jugador en entorno compartido pide el mejor movimiento conocido para su equipo en un estado concreto según la política generada por los jugadores en entorno exclusivo.		Movimiento

Tabla 73 : Interacción Pedir Mejor Movimiento en Entorno Exclusivo

<b>Pedir Política Compartida</b>		
<b>Jugador en entorno compartido</b>	<b>Gestor de conocimiento en entorno Compartido</b>	Agente, equipo
Un agente jugador en entorno compartido pide conocer la política que describe los mejores movimientos conocidos para su equipo.		Política colectiva

Tabla 74 : Interacción Pedir Política Compartida

#### II.2.4. Interacciones con la organización de Jugadores en Entorno Exclusivo

<b>Empezar Entrenamiento Exclusivo</b>		
<b>Control del entrenamiento en entorno exclusivo</b>	<b>Jugador en Entorno Exclusivo</b>	Agente
Control del entrenamiento en entorno exclusivo informa a un jugador de que puede empezar a ejecutar su proceso de aprendizaje.		Confirmación

Tabla 75 : Interacción Empezar Entrenamiento Exclusivo

#### II.2.5. Interacciones con la organización de Jugadores en Entorno Compartido

<b>Empezar Episodio de Entrenamiento Compartido</b>		
<b>Control del entrenamiento en entorno Compartido</b>	<b>Jugador en Entorno Compartido</b>	Agente
Control del entrenamiento en entorno Compartido informa a un jugador en entorno compartido de que puede empezar a ejecutar un episodio de su proceso de aprendizaje.		Confirmación

Tabla 76 : Interacción Empezar Episodio de Entrenamiento Compartido

<b>Empezar Ciclo de Entrenamiento Compartido</b>		
<b>Control del entrenamiento en entorno Compartido</b>	<b>Jugador en Entorno Compartido</b>	Agente
Control del entrenamiento en entorno Compartido informa a un jugador en entorno compartido de que puede empezar a ejecutar un ciclo de su proceso de aprendizaje.		Confirmación

Tabla 77 : Interacción Empezar Ciclo de Entrenamiento Compartido

## II.2.6. Interacciones con la organización de control

<b>Listo para Entrenamiento Exclusivo</b>		
<b>Jugador en Entorno Exclusivo</b>	<b>Control del entrenamiento en Entorno Exclusivo</b>	Agente
Un jugador en entorno exclusivo informa a Control del entrenamiento en entorno exclusivo de que se encuentra listo para ejecutar su proceso de aprendizaje		Confirmación

Tabla 78 : Interacción Listo para Entrenamiento Exclusivo

<b>Entrenamiento Exclusivo Acabado</b>		
<b>Jugador en Entorno Exclusivo</b>	<b>Control del entrenamiento en Entorno Exclusivo</b>	Agente
Un jugador en entorno exclusivo informa a control del entrenamiento en entorno exclusivo de que ha acabado su proceso de aprendizaje.		Confirmación

Tabla 79 : Interacción Entrenamiento Exclusivo Acabado

<b>Listo para Episodio de Entrenamiento Compartido</b>		
<b>Jugador en Entorno Compartido</b>	<b>Control del entrenamiento en Entorno Compartido</b>	Agente
Un jugador en entorno compartido informa a Control del entrenamiento en entorno compartido de que se encuentra preparado para ejecutar un episodio de su proceso de aprendizaje.		Confirmación

Tabla 80 : Interacción Listo para Episodio de Entrenamiento Compartido

<b>Acabado Episodio de Entrenamiento Compartido</b>		
<b>Jugador en Entorno Compartido</b>	<b>Control del entrenamiento en Entorno Compartido</b>	Agente
Un jugador en entorno compartido informa a Control del entrenamiento en entorno compartido de que ha acabado un episodio de su proceso de aprendizaje.		Confirmación

Tabla 81 : Interacción Acabado Episodio de Entrenamiento Compartido

<b>Listo para Ciclo de Entrenamiento Compartido</b>		
<b>Jugador en Entorno Compartido</b>	<b>Control del entrenamiento en Entorno Compartido</b>	Agente
Un jugador en entorno compartido informa a Control del entrenamiento compartido de que se encuentra preparado para ejecutar un ciclo de su proceso de aprendizaje.		Confirmación

Tabla 82 : Interacción Listo para Ciclo de Entrenamiento Compartido

<b>Acabado Ciclo de Entrenamiento en Entorno Compartido</b>		
<b>Jugador en Entorno Compartido</b>	<b>Control del entrenamiento en Entorno Compartido</b>	Agente
Un jugador en entorno compartido informa a Control del entrenamiento en entorno compartido de que ha acabado un ciclo de su proceso de aprendizaje.		Confirmación

Tabla 83 : Interacción Acabado Ciclo de Entrenamiento en Entorno Compartido

<b>Entrenamiento en Entorno Compartido Acabado</b>		
<b>Jugador en Entorno Compartido</b>	<b>Control del entrenamiento en Entorno Compartido</b>	Agente
Un jugador en entorno compartido informa a Control del entrenamiento en entorno compartido de que ha acabado su proceso de aprendizaje.		Confirmación

Tabla 84 : Interacción Entrenamiento en Entorno Compartido Acabado

<b>Pedir Registro en Entorno Tablero</b>		
<b>Jugador en Entorno Compartido o Jugador en Entorno Exclusivo</b>	<b>Control del entrenamiento en Entorno Compartido o Control del entrenamiento en Entorno Exclusivo</b>	Agente
Un agente jugador solicita ser registrado en el entorno Tablero, en el que ejecutará su proceso de aprendizaje.		Agente Registrado o Agente No Registrado

Tabla 85 : Interacción Pedir Registro en Entorno Tablero

<b>Informar Entrenamiento en Entorno Exclusivo Acabado</b>		
<b>Control del entrenamiento en entorno exclusivo</b>	<b>Control del entrenamiento en entorno compartido</b>	Agente
Se informa de que el entrenamiento en entorno exclusivo ha acabado y puede empezar el entrenamiento en entorno compartido.		Confirmación

Tabla 86 : Interacción Informar Entrenamiento en Entorno Exclusivo Acabado



## II.2.7. Interacciones con los roles arquitectónicos

<b>Solicitud de Uso</b>		
<b>Jugador en Entorno Compartido o Jugador en Entorno Exclusivo</b>	<b>Servidor</b>	Agente
Un agente Jugador solicita permiso para usar los servicios de alguno de los servidores presentes en el sistema.		Agente Autorizado o Agente No Autorizado

Tabla 87 : Interacción Solicitud de Uso

<b>Pedir Registro</b>		
<b>Registrador DF</b>	<b>Agente DF</b>	Agente
Un agente solicita ser registrado en el agente DF de la plataforma JADE.		Confirmación

Tabla 88 : Interacción Pedir Registro

<b>Encontrar Agente</b>		
<b>Buscador agentes</b>	<b>Agente DF</b>	Descripción agente
Un agente jugador solicita al agente DF que le informe acerca de todos los agentes conocidos que coinciden con la descripción dada.		Lista de agentes

Tabla 89 : Interacción Encontrar Agente

<b>Finalizar Ejecución</b>		
<b>Control del entrenamiento en Entorno Compartido o Control del entrenamiento en Entorno Exclusivo</b>	<b>Finalizable</b>	Agente
Alguno de los controles solicita a un agente finalizable que cese su ejecución.		Ejecución finalizada o ejecución no finalizada

Tabla 90 : Interacción Finalizar Ejecución



## Anexo III: Comportamientos asociados a los agentes

### III.1. Comportamientos asociados al agente estado

<b>Comportamiento</b>	Registro en DF	<b>Heredado del Rol</b>	RegistradorDF
<b>Función</b>	Registra los servicios ofrecidos por el Agente Estado en el agente DF de la plataforma JADE		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Servidor	<b>Heredado del Rol</b>	Servidor
<b>Función</b>	Permite que los agentes soliciten el uso de los servicios del agente.		
<b>Condiciones</b>	Los usuarios sólo pueden ser Agente Jugador Entorno Exclusivo y Agente Jugador Entorno Compartido		
<b>Comportamiento</b>	Finalizar	<b>Heredado del Rol</b>	Finalizable
<b>Función</b>	Permite que otros agentes puedan finalizar la ejecución del agente		
<b>Condiciones</b>	Sólo podrán finalizar la ejecución del agente Control Entrenamiento Entorno Exclusivo y Control Entrenamiento Entorno Compartido		
<b>Comportamiento</b>	Gestor Estado Exclusivo	<b>Heredado del Rol</b>	Gestor Estado
<b>Función</b>	Permite que otros agentes en entorno exclusivo pidan y reciban el estado en el que se encuentran en el Entorno Tablero		
<b>Condiciones</b>	Sólo podrán usar estas operaciones los agentes que previamente hayan solicitado permiso mediante el rol Servidor del mismo agente y sean de tipo Agente Jugador en Entorno Exclusivo		
<b>Comportamiento</b>	Gestor Estado Compartido	<b>Heredado del Rol</b>	Gestor Estado
<b>Función</b>	Permite que otros agentes en entorno compartido pidan y reciban el estado en el que se encuentran en el Entorno Tablero		
<b>Condiciones</b>	Sólo podrán usar estas operaciones los agentes que previamente hayan solicitado permiso mediante el rol Servidor del mismo agente y sean de tipo Agente Jugador en Entorno Compartido		

Tabla 91 : Comportamientos asociados al agente estado

### III.2. Comportamientos asociados al agente movimiento

<b>Comportamiento</b>	Registro en DF	<b>Heredado del Rol</b>	RegistradorDF
<b>Función</b>	Registra los servicios ofrecidos por el Agente Movimiento en el agente DF de la plataforma JADE		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Servidor	<b>Heredado del Rol</b>	Servidor
<b>Función</b>	Permite que los agentes soliciten el uso de los servicios del agente.		
<b>Condiciones</b>	Los usuarios sólo pueden ser Agente Jugador Entorno Exclusivo y Agente Jugador Entorno Compartido		
<b>Comportamiento</b>	Finalizar	<b>Heredado del Rol</b>	Finalizable
<b>Función</b>	Permite que otros agentes puedan finalizar la ejecución del agente		
<b>Condiciones</b>	Sólo podrán finalizar la ejecución del agente Control Entrenamiento		

	Entorno Exclusivo y Control Entrenamiento Entorno Compartido		
<b>Comportamiento</b>	Gestor Movimiento Entorno Exclusivo	<b>Heredado del Rol</b>	Gestor Movimiento
<b>Función</b>	Permite que agentes Jugador en Entorno Exclusivo puedan ejecutar movimientos en el entorno Tablero, siendo informados de su refuerzo.		
<b>Condiciones</b>	Sólo podrán usar estas operaciones los agentes que previamente hayan solicitado permiso mediante el rol Servidor del mismo agente y sean de tipo Agente Jugador en Entorno Exclusivo		
<b>Comportamiento</b>	Gestor Movimiento Entorno Compartido	<b>Heredado del Rol</b>	Gestor Estado
<b>Función</b>	Permite que agentes Jugador en Entorno Compartido puedan ejecutar movimientos en el entorno Tablero, siendo informados de su refuerzo.		
<b>Condiciones</b>	Sólo podrán usar estas operaciones los agentes que previamente hayan solicitado permiso mediante el rol Servidor del mismo agente y sean de tipo Agente Jugador en Entorno Compartido		

Tabla 92 : Comportamientos asociados al agente movimiento

### III.3. Comportamientos asociados al agente reinicio

<b>Comportamiento</b>	Registro en DF	<b>Heredado del Rol</b>	RegistradorDF
<b>Función</b>	Registra los servicios ofrecidos por el Agente Reinicio en el agente DF de la plataforma JADE		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Servidor	<b>Heredado del Rol</b>	Servidor
<b>Función</b>	Permite que los agentes soliciten el uso de los servicios del agente.		
<b>Condiciones</b>	Los usuarios sólo pueden ser Agente Jugador Entorno Exclusivo y Agente Jugador Entorno Compartido.		
<b>Comportamiento</b>	Finalizar	<b>Heredado del Rol</b>	Finalizable
<b>Función</b>	Permite que otros agentes puedan finalizar la ejecución del agente		
<b>Condiciones</b>	Sólo podrán finalizar la ejecución del agente Control Entrenamiento Entorno Exclusivo y Control Entrenamiento Entorno Compartido		
<b>Comportamiento</b>	Gestor Reinicio	<b>Heredado del Rol</b>	Gestor Reinicio
<b>Función</b>	Permite que los agentes puedan reiniciar su posición en el entorno Tablero		
<b>Condiciones</b>	Sólo podrán usar estas operaciones los agentes que previamente hayan solicitado permiso mediante el rol Servidor del mismo agente y que sean Agente Jugador Entorno Exclusivo y Agente Jugador Entorno Compartido.		

Tabla 93 : Comportamientos asociados al agente reinicio

### III.4. Comportamientos asociados al agente gestor estadístico

<b>Comportamiento</b>	Registro en DF	<b>Heredado del Rol</b>	RegistradorDF
<b>Función</b>	Registra los servicios ofrecidos por el Agente Gestor Estadístico en el agente DF de la plataforma JADE		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Servidor	<b>Heredado del Rol</b>	Servidor
<b>Función</b>	Permite que los agentes soliciten el uso de los servicios del agente.		

<b>Condiciones</b>	Los usuarios sólo pueden ser Agente Jugador Entorno Exclusivo y Agente Jugador Entorno Compartido.		
<b>Comportamiento</b>	Finalizar	<b>Heredado del Rol</b>	Finalizable
<b>Función</b>	Permite que otros agentes puedan finalizar la ejecución del agente		
<b>Condiciones</b>	Sólo podrán finalizar la ejecución del agente Control Entrenamiento Entorno Exclusivo y Control Entrenamiento Entorno Compartido		
<b>Comportamiento</b>	Gestor Estadístico Estadísticas Entorno Exclusivo	<b>Heredado del Rol</b>	Gestor Estadístico
<b>Función</b>	Recibe las estadísticas generadas por los Agente Jugador Entorno Exclusivo.		
<b>Condiciones</b>	Sólo podrán usar estas operaciones los agentes que previamente hayan solicitado permiso mediante el rol Servidor del mismo agente y que sean Agente Jugador Entorno Exclusivo.		
<b>Comportamiento</b>	Gestor Estadístico Estadísticas Entorno Compartido	<b>Heredado del Rol</b>	Gestor Estadístico
<b>Función</b>	Recibe las estadísticas generadas por los Agente Jugador Entorno Compartido.		
<b>Condiciones</b>	Sólo podrán usar estas operaciones los agentes que previamente hayan solicitado permiso mediante el rol Servidor del mismo agente y que sean Agente Jugador Entorno Compartido.		

Tabla 94 : Comportamientos asociados al agente gestor estadístico

### III.5. Comportamientos asociados al agente procesador estadístico

<b>Comportamiento</b>	Registro en DF	<b>Heredado del Rol</b>	RegistradorDF
<b>Función</b>	Registra los servicios ofrecidos por el Agente Procesador Estadístico en el agente DF de la plataforma JADE		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Servidor	<b>Heredado del Rol</b>	Servidor
<b>Función</b>	Permite que los agentes soliciten el uso de los servicios del agente.		
<b>Condiciones</b>	Los usuarios sólo pueden ser Control Entrenamiento Entorno Compartido.		
<b>Comportamiento</b>	Finalizar	<b>Heredado del Rol</b>	Finalizable
<b>Función</b>	Permite que otros agentes puedan finalizar la ejecución del agente		
<b>Condiciones</b>	Sólo podrán finalizar la ejecución del agente Control Entrenamiento Entorno Exclusivo y Control Entrenamiento Entorno Compartido		
<b>Comportamiento</b>	Procesador	<b>Heredado del Rol</b>	Procesador
<b>Función</b>	Ejecuta periódicamente el procesamiento de estadísticas.		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Procesador Estadístico	<b>Heredado del Rol</b>	Procesador Estadístico
<b>Función</b>	Recibe peticiones de actualización de estadísticas para mostrarlas al usuario por pantalla.		
<b>Condiciones</b>	Sólo podrán usar estas operaciones los agentes que previamente hayan solicitado permiso mediante el rol Servidor del mismo agente y que sean Control Entrenamiento Entorno Compartido.		

Tabla 95 : Comportamientos asociados al agente procesador estadístico

### III.6. Comportamientos asociados al agente gestor de conocimiento exclusivo

<b>Comportamiento</b>	Registro en DF	<b>Heredado del Rol</b>	RegistradorDF
<b>Función</b>	Registra los servicios ofrecidos por el Agente Gestor Conocimiento Exclusivo en el agente DF de la plataforma JADE		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Servidor	<b>Heredado del Rol</b>	Servidor
<b>Función</b>	Permite que los agentes soliciten el uso de los servicios del agente.		
<b>Condiciones</b>	Los usuarios sólo pueden ser Agente Jugador Entorno Compartido o Agente Jugador Entorno Exclusivo.		
<b>Comportamiento</b>	Finalizar	<b>Heredado del Rol</b>	Finalizable
<b>Función</b>	Permite que otros agentes puedan finalizar la ejecución del agente		
<b>Condiciones</b>	Sólo podrán finalizar la ejecución del agente Control Entrenamiento Entorno Exclusivo y Control Entrenamiento Entorno Compartido		
<b>Comportamiento</b>	Inserción Conocimiento Exclusivo	<b>Heredado del Rol</b>	Gestor Conocimiento Exclusivo
<b>Función</b>	Recibe los datos de una política generada por algun Agente Jugador en Entorno Exclusivo		
<b>Condiciones</b>	Sólo podrán usar estas operaciones los agentes que previamente hayan solicitado permiso mediante el rol Servidor del mismo agente y que sean Agente Jugador Entorno Exclusivo.		
<b>Comportamiento</b>	Lectura Conocimiento Exclusivo	<b>Heredado del Rol</b>	Gestor Conocimiento Exclusivo
<b>Función</b>	Envía los datos de la política compadida al Agente Jugador en Entorno Compartido que lo haya solicitado		
<b>Condiciones</b>	Sólo podrán usar estas operaciones los agentes que previamente hayan solicitado permiso mediante el rol Servidor del mismo agente y que sean Agente Jugador Entorno Compartido.		

Tabla 96 : Comportamientos asociados al agente gestor conocimiento exclusivo

### III.7. Comportamientos asociados al agente gestor de conocimiento compartido

<b>Comportamiento</b>	Registro en DF	<b>Heredado del Rol</b>	RegistradorDF
<b>Función</b>	Registra los servicios ofrecidos por el Agente Gestor Conocimiento Compartido en el agente DF de la plataforma JADE		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Servidor	<b>Heredado del Rol</b>	Servidor
<b>Función</b>	Permite que los agentes soliciten el uso de los servicios del agente.		
<b>Condiciones</b>	Los usuarios sólo pueden ser Agente Jugador Entorno Compartido.		
<b>Comportamiento</b>	Finalizar	Heredado del Rol	Finalizable
<b>Función</b>	Permite que otros agentes puedan finalizar la ejecución del agente		
<b>Condiciones</b>	Sólo podrán finalizar la ejecución del agente Control Entrenamiento Entorno Exclusivo y Control Entrenamiento Entorno Compartido		

<b>Comportamiento</b>	Inserción Conocimiento Compartido	<b>Heredado del Rol</b>	Gestor Conocimiento Compartido
<b>Función</b>	Recibe los datos de una política generada por algun Agente Jugador en Entorno Compartido		
<b>Condiciones</b>	Sólo podrán usar estas operaciones los agentes que previamente hayan solicitado permiso mediante el rol Servidor del mismo agente y que sean Agente Jugador Entorno Compartido.		
<b>Comportamiento</b>	Lectura Conocimiento Compartido	<b>Heredado del Rol</b>	Gestor Conocimiento Compartido
<b>Función</b>	Envía los datos de la política compartida al Agente Jugador en Entorno Compartido que lo haya solicitado		
<b>Condiciones</b>	Sólo podrán usar estas operaciones los agentes que previamente hayan solicitado permiso mediante el rol Servidor del mismo agente y que sean Agente Jugador Entorno Compartido.		

Tabla 97 : Comportamientos asociados al agente gestor conocimiento compartido

### III.8. Comportamientos asociados al agente procesador de conocimiento

<b>Comportamiento</b>	Registro en DF	<b>Heredado del Rol</b>	RegistradorDF
<b>Función</b>	Registra los servicios ofrecidos por el Agente Procesador Conocimiento en el agente DF de la plataforma JADE		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Finalizar	<b>Heredado del Rol</b>	Finalizable
<b>Función</b>	Permite que otros agentes puedan finalizar la ejecución del agente		
<b>Condiciones</b>	Sólo podrán finalizar la ejecución del agente Control Entrenamiento Entorno Exclusivo y Control Entrenamiento Entorno Compartido		
<b>Comportamiento</b>	Procesador	<b>Heredado del Rol</b>	Procesador
<b>Función</b>	Ejecuta periódicamente la tarea de procesamiento de conocimiento.		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Procesador Conocimiento	<b>Heredado del Rol</b>	Procesador Estadístico
<b>Función</b>	Ejecuta la tarea de procesamiento de conocimiento.		
<b>Condiciones</b>	Ninguna		

Tabla 98 : Comportamientos asociados al agente procesador de conocimiento

### III.9. Comportamientos asociados al agente jugador en entorno exclusivo

<b>Comportamiento</b>	Registro en DF	<b>Heredado del Rol</b>	RegistradorDF
<b>Función</b>	Registrar al agente Jugador en Entorno Exclusivo en el agente DF de JADE, para que pueda ser encontrado por otros agentes		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Finalizar	<b>Heredado del Rol</b>	Finalizable
<b>Función</b>	Permite que otros agentes puedan finalizar la ejecución del agente		

<b>Condiciones</b>	Sólo podrán finalizar la ejecución del agente Control Entrenamiento Entorno Exclusivo y Control Entrenamiento Entorno Compartido		
<b>Comportamiento</b>	Buscar Agente	<b>Heredado del Rol</b>	Buscador Agente
<b>Función</b>	Busca a los agentes necesarios para su ejecución.		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Registro en entorno Tablero	<b>Heredado del Rol</b>	Jugador en Entorno Exclusivo
<b>Función</b>	Registra al agente en el entorno Tablero.		
<b>Condiciones</b>	Haber solicitado el uso de los servicios de Control del Entrenamiento en Entorno Exclusivo previamente y haber sido autorizado para ello.		
<b>Comportamiento</b>	Entrenamiento en Entorno Exclusivo	<b>Heredado del Rol</b>	Jugador en Entorno Exclusivo
<b>Función</b>	Ejecuta la rutina de entrenamiento del agente.		
<b>Condiciones</b>	Haber solicitado el uso de los servicios de Control del Entrenamiento en Entorno Exclusivo previamente, Agente Estado, Agente Movimiento y Agente Reinicio y haber sido autorizado por todos ellos.		
<b>Comportamiento</b>	Enviar Conocimiento	Heredado del Rol	Jugador en Entorno Exclusivo
<b>Función</b>	Envía el conocimiento adquirido durante el proceso de aprendizaje.		
<b>Condiciones</b>	Haber terminado la rutina de entrenamiento. Haber solicitado el uso de los servicios de Agente Gestor de Conocimiento Entorno Exclusivo y haber sido autorizado para ello.		

Tabla 99 : Comportamientos asociados al agente jugador en entorno exclusivo

### III.10. Comportamientos asociados al agente jugador en entorno compartido

<b>Comportamiento</b>	Registro en DF	<b>Heredado del Rol</b>	RegistradorDF
<b>Función</b>	Registrar al agente Jugador en Entorno Compartido en el agente DF de JADE, para que pueda ser encontrado por otros agentes		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Finalizar	<b>Heredado del Rol</b>	Finalizable
<b>Función</b>	Permite que otros agentes puedan finalizar la ejecución del agente		
<b>Condiciones</b>	Sólo podrán finalizar la ejecución del agente Control Entrenamiento Entorno Exclusivo y Control Entrenamiento Entorno Compartido		
<b>Comportamiento</b>	Buscar Agente	<b>Heredado del Rol</b>	Buscador Agente
<b>Función</b>	Busca a los agentes necesarios para su ejecución.		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Registro en entorno Tablero	<b>Heredado del Rol</b>	Jugador en Entorno Compartido
<b>Función</b>	Registra al agente en el entorno Tablero.		
<b>Condiciones</b>	Haber solicitado el uso de los servicios de Control del Entrenamiento en Entorno Compartido previamente y haber sido autorizado para ello.		
<b>Comportamiento</b>	Entrenamiento en Entorno Exclusivo	<b>Heredado del Rol</b>	Jugador en Entorno Compartido
<b>Función</b>	Ejecuta la rutina de entrenamiento del agente.		
<b>Condiciones</b>	Haber solicitado el uso de los servicios de Control del Entrenamiento en Entorno Compartido previamente, Agente Estado, Agente Movimiento y		



	Agente Reinicio y haber sido autorizado por todos ellos.
--	--

Tabla 100 : Comportamientos asociados al agente jugador en entorno compartido

### III.11. Comportamientos asociados al agente control del entrenamiento en entorno exclusivo

<b>Comportamiento</b>	Registro en DF	<b>Heredado del Rol</b>	RegistradorDF
<b>Función</b>	Registrar al agente Control del entrenamiento en Entorno Exclusivo en el agente DF de JADE, para que pueda ser encontrado por otros agentes		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Finalizar	<b>Heredado del Rol</b>	Finalizable
<b>Función</b>	Permite que otros agentes puedan finalizar la ejecución del agente		
<b>Condiciones</b>	Sólo podrán finalizar la ejecución agentes Control Entrenamiento Entorno Compartido.		
<b>Comportamiento</b>	Buscar Agente	<b>Heredado del Rol</b>	Buscador Agente
<b>Función</b>	Busca a los agentes necesarios para su ejecución.		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Servidor	<b>Heredado del Rol</b>	Servidor
<b>Función</b>	Permite que los agentes soliciten el uso de los servicios del agente.		
<b>Condiciones</b>	Los usuarios sólo pueden ser Agente Jugador Entorno Exclusivo.		
<b>Comportamiento</b>	Registro en entorno Tablero	<b>Heredado del Rol</b>	Control del entrenamiento en Entorno Exclusivo
<b>Función</b>	Permite que los agentes soliciten el registro en el entorno Tablero.		
<b>Condiciones</b>	Los usuarios sólo pueden ser Agente Jugador Entorno Exclusivo.		
<b>Comportamiento</b>	Control Entrenamiento en Entorno Exclusivo	<b>Heredado del Rol</b>	Control del entrenamiento en Entorno Exclusivo
<b>Función</b>	Controla ordenadamente la ejecución del entrenamiento en entorno exclusivo.		
<b>Condiciones</b>	Haber recibido las peticiones de entrenamiento en entorno exclusivo necesarias.		
<b>Comportamiento</b>	Finalizar Entrenamiento en Entorno Exclusivo	<b>Heredado del Rol</b>	Control del entrenamiento en Entorno Exclusivo
<b>Función</b>	Finaliza a los agentes implicados en el entrenamiento en entorno exclusivo e informa a Control del entrenamiento en Entorno Compartido de que el entrenamiento en entorno exclusivo ha acabado		
<b>Condiciones</b>	Haber acabado el entrenamiento. Haber contactado con Control del entrenamiento en Entorno Compartido.		

Tabla 101 : Comportamientos asociados al agente control del entrenamiento en entorno exclusivo

### III.12. Comportamientos asociados al agente control del entrenamiento en entorno compartido

<b>Comportamiento</b>	Registro en DF	<b>Heredado del Rol</b>	RegistradorDF
<b>Función</b>	Registrar al agente Control del entrenamiento en Entorno Compartido en		

	el agente DF de JADE, para que pueda ser encontrado por otros agentes		
<b>Condiciones</b>	Ninguna		
<b>Comportamiento</b>	Servidor	<b>Heredado del Rol</b>	Servidor
<b>Función</b>	Permite que los agentes soliciten el uso de los servicios del agente.		
<b>Condiciones</b>	Los usuarios sólo pueden ser Agente Jugador Entorno Compartido.		
<b>Comportamiento</b>	Registro en entorno Tablero	<b>Heredado del Rol</b>	Control del enternamiento en Entorno Compartido
<b>Función</b>	Permite que los agentes soliciten el registro en el entorno Tablero.		
<b>Condiciones</b>	Los usuarios sólo pueden ser Agente Jugador Entorno Compartido.		
<b>Comportamiento</b>	Control Entrenamiento en Entorno Compartido	<b>Heredado del Rol</b>	Control del enternamiento en Entorno Compartido
<b>Función</b>	Controla ordenadamente la ejecución del entrenamiento en entorno exclusivo.		
<b>Condiciones</b>	Haber recibido las peticiones de entrenamiento en entorno compartido necesarias.		
<b>Comportamiento</b>	Control Episodio Entrenamiento en Entorno Compartido	<b>Heredado del Rol</b>	Control del enternamiento en Entorno Compartido
<b>Función</b>	Controla ordenadamente la ejecución de los episodios de entrenamiento en entorno exclusivo.		
<b>Condiciones</b>	Haber recibido las peticiones para ejecutar un episodio de entrenamiento en entorno compartido necesarias.		
<b>Comportamiento</b>	Control Ciclo Entrenamiento en Entorno Compartido	<b>Heredado del Rol</b>	Control del enternamiento en Entorno Compartido
<b>Función</b>	Controla ordenadamente la ejecución de los ciclos de entrenamiento en entorno exclusivo.		
<b>Condiciones</b>	Haber recibido las peticiones para ejecutar un ciclo de entrenamiento en entorno compartido necesarias.		
<b>Comportamiento</b>	Finalizar Entrenamiento en Entorno Compartido	<b>Heredado del Rol</b>	Control del enternamiento en Entorno Compartido
<b>Función</b>	Finaliza a los agentes implicados en el entrenamiento en entorno compartido .		
<b>Condiciones</b>	Haber acabado el entrenamiento.		

Tabla 102 : Comportamientos asociados al agente control delentrenamiento en entorno compartido

## Anexo IV: Parámetros configurables de los agentes

### IV.1. Parámetros configurables del agente estado

<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
Clientes	Entero	Especifica el número máximo de clientes que puede tener el agente en su rol Servidor.

Tabla 103 : Parámetros configurables del agente estado

### IV.2. Parámetros configurables del agente movimiento

<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
Clientes	Entero	Especifica el número máximo de clientes que puede tener el agente en su rol Servidor.
Movimientos autorizados en Entrenamiento en Entorno Exclusivo.	Movimiento[]	Lista de movimientos que los agentes Jugador en Entorno Exclusivo están autorizados a realizar en su proceso de aprendizaje.
Movimientos autorizados en Entrenamiento en Entorno Compartido.	Movimiento[]	Lista de movimientos que los agentes Jugador en Entorno Compartido están autorizados a realizar en su proceso de aprendizaje.

Tabla 104 : Parámetros configurables del agente movimiento

### IV.3. Parámetros configurables del agente reinicio

<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
Clientes	Entero	Especifica el número máximo de clientes que puede tener el agente en su rol Servidor.

Tabla 105 : Parámetros configurables del agente reinicio

### IV.4. Parámetros configurables del agente gestor estadístico

<b>Parámetro</b>	<b>Tipo</b>	<b>Descripción</b>
Clientes	Entero	Especifica el número máximo de clientes que puede tener el agente en su rol Servidor.

Tabla 106 : Parámetros configurables del agente gestor estadístico

#### IV.5. Parámetros configurables del agente procesador estadístico

Parámetro	Tipo	Descripción
Clientes	Entero	Especifica el número máximo de clientes que puede tener el agente en su rol Servidor.

Tabla 107 : Parámetros configurables del agente procesador estadístico

#### IV.6. Parámetros configurables del agente gestor de conocimiento exclusivo

Parámetro	Tipo	Descripción
Clientes	Entero	Especifica el número máximo de clientes que puede tener el agente en su rol Servidor.

Tabla 108 : Parámetros configurables del agente gestor de conocimiento exclusivo

#### IV.7. Parámetros configurables del agente gestor de conocimiento compartido

Parámetro	Tipo	Descripción
Clientes	Entero	Especifica el número máximo de clientes que puede tener el agente en su rol Servidor.

Tabla 109 : Parámetros configurables del agente gestor de conocimiento compartido

#### IV.8. Parámetros configurables del agente jugador en entorno exclusivo

Parámetro	Tipo	Descripción
Episodios	Entero	Número de episodios que durará el proceso de aprendizaje.
Ciclos	Entero	Número de ciclos que durará el proceso de aprendizaje.
$\gamma$	Real	Parámetro de descuento del algoritmo de Q-Learning.
Nulos	Booleano	Indica si se podrá usar el movimiento nulo durante el proceso de aprendizaje.
Refuerzos Negativos	Booleano	Indica si se podrán usar refuerzos negativos durante el proceso de aprendizaje.
$\epsilon$	Real	Parámetro $\epsilon$ de la política de decisión de movimiento $\epsilon$ -greedy.
Equipo	Entero	Identificador del equipo al que pertenece.

Tabla 110 : Parámetros configurables del agente Jugador en Entorno Exclusivo

## IV.8. Parámetros configurables del agente jugador en entorno compartido

<i>Parámetro</i>	<i>Tipo</i>	<i>Descripción</i>
Episodios	Entero	Número de episodios que durará el proceso de aprendizaje.
Ciclos	Entero	Número de ciclos que durará el proceso de aprendizaje.
$\gamma$	Real	Parámetro de descuento del algoritmo de Q-Learning.
$\alpha$	Real	Parámetro de enfriamiento del algoritmo de Q-Learning.
Refuerzos Negativos	Booleano	Indica si se podrán usar refuerzos negativos durante el proceso de aprendizaje.
Nulos	Booleano	Indica si se podrá usar el movimiento nulo durante el proceso de aprendizaje.
Equipo	Entero	Identificador del equipo al que pertenece.
Retardo	Entero	Número de episodios que se retardará el inicio del proceso de aprendizaje.
Porcentaje acciones aleatorias	Real	Parámetro que indica el porcentaje inicial de acciones aleatorias que se tomarán en la política de decisión de la acción a tomar.
Incremento porcentaje acciones aleatorias	Real	Incremento del porcentaje de acciones aleatorias que se sumará al finalizar los episodios en los que el agente llegue a su objetivo.
Mínimo porcentaje acciones aleatorias	Real	Mínimo porcentaje de acciones aleatorias al que se puede llegar.
Máximo porcentaje acciones aleatorias	Real	Mínimo porcentaje de acciones aleatorias al que se puede llegar.
Porcentaje acciones según la política exclusiva	Real	Parámetro que indica el porcentaje inicial de acciones según la política exclusiva que se tomarán en la política de decisión de la acción a tomar.
Incremento porcentaje acciones según la política exclusiva	Real	Incremento del porcentaje de acciones según la política exclusiva que se sumará al finalizar los episodios en los que el agente llegue a su objetivo.
Mínimo porcentaje acciones según la política exclusiva	Real	Mínimo porcentaje de acciones según la política exclusiva al que se puede llegar.
Máximo porcentaje acciones según la política exclusiva	Real	Mínimo porcentaje de acciones según la política exclusiva al que se puede llegar.
Porcentaje acciones según la política aprendida	Real	Parámetro que indica el porcentaje inicial de acciones según la política aprendida que se tomarán en la política de decisión de la acción a tomar.
Incremento porcentaje acciones según la política aprendida	Real	Incremento del porcentaje de acciones según la política aprendida que se sumará al finalizar los episodios en los que el agente llegue a su objetivo.
Mínimo porcentaje acciones según la política aprendida	Real	Mínimo porcentaje de acciones según la política aprendida al que se puede llegar.
Máximo porcentaje acciones según la	Real	Mínimo porcentaje de acciones según la política aprendida al que se puede llegar.

política aprendida		
Episodios para pedir conocimiento	Entero	Máximo número de episodios que se esperará para obtener la política compartida.
Ciclos para pedir conocimiento	Entero	Número de ciclos a partir del cual, si un episodio dura más, se obtendrá la política compartida.

**Tabla 111 : Parámetros configurables del agente Jugador en Entorno Compartido**

## Anexo V: Modelo de servicios

<b>Servicio</b>	<b>Obtener Estado</b>		
<b>Rol Iniciador</b>	Jugador en Entorno Exclusivo o Jugador en Entorno Compartido	<b>Rol servidor</b>	Gestor Estado
<b>Descripción</b>	Interacción mediante la cual un Agente Jugador en Entorno Exclusivo o un Agente Jugador en Entorno Compartido solicita y obtiene su estado actual.		
<b>Entrada</b>	Agente		
<b>Salida</b>	Estado o Estado Desconocido		
<b>Precondiciones</b>	El Agente que empieza la interacción debe haber solicitado el uso del Agente Estado con el que tramita la petición.		
<b>Postcondiciones</b>	El Agente conoce su posición dentro del entorno Tablero.		

Tabla 112 : Servicio Obtener Estado

<b>Servicio</b>	<b>Ejecutar Movimiento</b>		
<b>Rol Iniciador</b>	Jugador en Entorno Exclusivo o Jugador en Entorno Compartido	<b>Rol servidor</b>	Gestor Movimiento
<b>Descripción</b>	Interacción mediante la cual un Agente Jugador en Entorno Exclusivo o un Agente Jugador en Entorno Compartido solicita un movimiento dentro del entorno Tablero y recibe el refuerzo asociado a su estado y la acción ejecutada.		
<b>Entrada</b>	Agente, Movimiento.		
<b>Salida</b>	Refuerzo		
<b>Precondiciones</b>	El Agente que empieza la interacción debe haber solicitado el uso del Agente Movimiento con el que tramita la petición.		
<b>Postcondiciones</b>	El agente conoce el refuerzo asociado a su acción y estado.		

Tabla 113 : Servicio Ejecutar Movimiento

<b>Servicio</b>	<b>Pedir Reinicio</b>		
<b>Rol Iniciador</b>	Jugador en Entorno Exclusivo o Jugador en Entorno Compartido	<b>Rol servidor</b>	Gestor Reinicio
<b>Descripción</b>	Interacción mediante la cual un Agente Jugador en Entorno Exclusivo o un Agente Jugador en Entorno Compartido solicita que su estado sea reiniciado en el entorno Tablero		
<b>Entrada</b>	Agente, Equipo		
<b>Salida</b>	Confirmación / Rechazo		
<b>Precondiciones</b>	El Agente que empieza la interacción debe haber solicitado el uso del Agente Reinicio con el que tramita la petición.		
<b>Postcondiciones</b>	La posición del agente es reiniciada en el entorno Tablero.		

Tabla 114 : Servicio Pedir Reinicio

<b>Servicio</b>	<b>Enviar estadísticas entorno Exclusivo</b>		
<b>Rol Iniciador</b>	Jugador en Entorno Exclusivo	<b>Rol servidor</b>	Gestor estadísticas
<b>Descripción</b>	Interacción mediante la cual un Agente Jugador en Entorno envía la información estadística acerca de su proceso de aprendizaje al Gestor de estadísticas		
<b>Entrada</b>	Agente, Estadísticas en entorno exclusivo		
<b>Salida</b>	Confirmación / Rechazo		
<b>Precondiciones</b>	El agente que inicia la interacción debe haber acabado un ciclo de entrenamiento.		
<b>Postcondiciones</b>			

Tabla 115 : Servicio Enviar estadísticas entorno Exclusivo

<b>Servicio</b>	<b>Enviar estadísticas entorno Compartido</b>		
<b>Rol Iniciador</b>	Jugador en Entorno Compartido	Rol servidor	Gestor estadísticas
<b>Descripción</b>	Interacción mediante la cual un Agente Jugador en Entorno Compartido envía la información estadística acerca de su proceso de aprendizaje al Gestor de estadísticas		
<b>Entrada</b>	Agente, Estadísticas en entorno compartido		
<b>Salida</b>	Confirmación / Rechazo		
<b>Precondiciones</b>	El agente que inicia la interacción debe haber acabado un ciclo de entrenamiento.		
<b>Postcondiciones</b>			

Tabla 116 : Servicio Enviar estadísticas entorno Compartido

<b>Servicio</b>	<b>Actualizar gráfica</b>		
<b>Rol Iniciador</b>	Control del entrenamiento en entorno compartido	<b>Rol servidor</b>	Procesador estadístico
<b>Descripción</b>	Interacción mediante la cual el control del entrenamiento en entorno compartido pide que se actualice la gráfica que enseña al usuario las estadísticas de los distintos procesos de aprendizaje.		
<b>Entrada</b>	Agente		
<b>Salida</b>	Confirmación / Rechazo		
<b>Precondiciones</b>	Control del entrenamiento en entorno compartido debe haber detectado que todos los agentes han acabado un episodio de entrenamiento en entorno compartido.		
<b>Postcondiciones</b>	La gráfica que enseña al usuario las estadísticas de los distintos procesos de aprendizaje es actualizada.		

Tabla 117 : Servicio Actualizar gráfica



<b>Servicio</b>	<b>Recibir política exclusiva</b>		
<b>Rol Iniciador</b>	Jugador en Entorno Exclusivo	<b>Rol servidor</b>	Gestor de Conocimiento en Entorno exclusivo
<b>Descripción</b>	Interacción mediante la cual el Jugador en Entorno Exclusivo envía el conocimiento adquirido sobre el entorno al Gestor de Conocimiento en Entorno Exclusivo		
<b>Entrada</b>	Agente, Política Entorno Exclusivo.		
<b>Salida</b>	Confirmación / Rechazo		
<b>Precondiciones</b>	El Jugador en Entorno Exclusivo debe haber acabado su proceso de aprendizaje.		
<b>Postcondiciones</b>	Se incluye nuevo conocimiento en el entorno Repositorio.		

Tabla 118 : Servicio Recibir política exclusiva

<b>Servicio</b>	<b>Recibir política compartida</b>		
<b>Rol Iniciador</b>	Jugador en Entorno Compartido	<b>Rol servidor</b>	Gestor de Conocimiento en Entorno Compartido
<b>Descripción</b>	Interacción mediante la cual el Jugador en Entorno Compartido envía el conocimiento adquirido sobre el entorno al Gestor de Conocimiento en Entorno Compartido		
<b>Entrada</b>	Agente, Política Entorno Compartido		
<b>Salida</b>	Confirmación / Rechazo		
<b>Precondiciones</b>	El Jugador en Entorno Compartido debe haber acabado un episodio de su proceso de aprendizaje.		
<b>Postcondiciones</b>	Se incluye nuevo conocimiento en el entorno Repositorio.		

Tabla 119 : Servicio Recibir política compartida

<b>Servicio</b>	<b>Obtener mejor movimiento en entorno exclusivo</b>		
<b>Rol Iniciador</b>	Jugador en Entorno Compartido	<b>Rol servidor</b>	Gestor de Conocimiento en Entorno Exclusivo
<b>Descripción</b>	Interacción mediante la cual el Jugador en Entorno Compartido solicita el mejor movimiento conocido según la política de movimientos en entorno exclusivo.		
<b>Entrada</b>	Agente, Equipo, Estado		
<b>Salida</b>	Movimiento		
<b>Precondiciones</b>	Ninguna.		
<b>Postcondiciones</b>	El Jugador en Entorno Compartido recibe el conocimiento solicitado.		

Tabla 120 : Servicio Obtener mejor movimiento en entorno exclusivo

<b>Servicio</b>	<b>Obtener política en entorno compartida</b>		
<b>Rol Iniciador</b>	Jugador en Entorno Compartido	<b>Rol servidor</b>	Gestor de Conocimiento en Entorno Compartido
<b>Descripción</b>	Interacción mediante la cual el Jugador en Entorno Compartido solicita la política de los mejores movimientos conocidos en entorno compartido.		
<b>Entrada</b>	Agente, Equipo, Estado		
<b>Salida</b>	Política		
<b>Precondiciones</b>	Ninguna.		
<b>Postcondiciones</b>	El Jugador en Entorno Compartido recibe el conocimiento solicitado.		

Tabla 121 : Servicio Obtener política en entorno compartida

<b>Servicio</b>	<b>Solicitud de uso</b>		
<b>Rol Iniciador</b>	Jugador en Entorno Exclusivo o Jugador en Entorno Compartido	<b>Rol servidor</b>	Servidor
<b>Descripción</b>	Interacción mediante la cual el Jugador en Entorno Exclusivo o Jugador en Entorno Compartido solicita autorización para usar los servicios de un Servidor.		
<b>Entrada</b>	Agente		
<b>Salida</b>	Confirmación / Rechazo		
<b>Precondiciones</b>	Se ha buscado el servidor en el agente DF de JADE.		
<b>Postcondiciones</b>	El jugador puede usar los servicios del servidor.		

Tabla 122 : Servicio Solicitud de uso

<b>Servicio</b>	<b>Pedir registro</b>		
<b>Rol Iniciador</b>	RegistradorDF	<b>Rol receptor</b>	Agente DF
<b>Descripción</b>	Interacción mediante la cual el Jugador en Entorno Exclusivo o Jugador en Entorno Compartido se registra en el agente DF de JADE.		
<b>Entrada</b>	Agente		
<b>Salida</b>	Confirmación / Rechazo		
<b>Precondiciones</b>	Ninguna		
<b>Postcondiciones</b>	El agente queda registrado en el agente DF de JADE.		

Tabla 123 : Servicio Pedir registro

<b>Servicio</b>	<b>Finalizar ejecución</b>		
<b>Rol Iniciador</b>	Control del Entrenamiento en Entorno Exclusivo o Control del Entrenamiento en Entorno Compartido	<b>Rol servidor</b>	Finalizable
<b>Descripción</b>	Interacción mediante la cual el Control del Entrenamiento en Entorno Exclusivo o Control del Entrenamiento en Entorno Compartido solicita a un agente que cese su ejecución.		
<b>Entrada</b>	Agente		
<b>Salida</b>	Confirmación / Rechazo		
<b>Precondiciones</b>	Ninguna		
<b>Postcondiciones</b>	El agente finaliza su ejecución		

Tabla 124 : Servicio Finalizar ejecución

## Anexo VI: Mensajes del sistema

A continuación se detallan todos los mensajes necesarios para la implementación del sistema. Según el estándar de la FIPA todos ellos tienen un tipo asociado para facilitar su tratamiento.

M1	
<b>PedirEstado</b>	
<b>Descripción</b>	El agente Jugador pide ser informado de su estado en el entorno Tablero
<b>Tipo</b>	REQUEST
<b>Emisor</b>	Jugador en Entorno Exclusivo o Jugador en Entorno Compartido
<b>Receptor</b>	Agente Estado
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que pide conocer su posición

Mensaje 1 : Pedir Estado

M2	
<b>EstadoExclusivoAgente</b>	
<b>Descripción</b>	Describe el estado de un agente en el entorno que disfruta en exclusiva.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Agente Estado
<b>Receptor</b>	Jugador en Entorno Exclusivo
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente del que se describe el estado.
Estado	Estado del agente.

Mensaje 2 : Estado Exclusivo Agente

M3	
<b>EstadoCompartidoAgente</b>	
<b>Descripción</b>	Describe el estado de un agente en el entorno que comparte con otros agentes.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Agente Estado
<b>Receptor</b>	Jugador en Entorno Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente del que se describe el estado.
Estado	Estado del agente.

Mensaje 3 : Estado Compartido Agente

**M4**

<b>EstadoExclusivoDesconocido</b>	
<b>Descripción</b>	Informa de que el estado de un agente es desconocido en un entorno exclusivo.
<b>Tipo</b>	REFUSE
<b>Emisor</b>	Agente Estado
<b>Receptor</b>	Jugador en Entorno Exclusivo
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente del que se desconoce el estado.
Motivo	Motivo por el que se desconoce el estado

**Mensaje 4 : Estado Exclusivo Desconocido**

**M5**

<b>EstadoCompartidoDesconocido</b>	
<b>Descripción</b>	Informa de que el estado de un agente es desconocido en un entorno exclusivo.
<b>Tipo</b>	REFUSE
<b>Emisor</b>	Entorno
<b>Receptor</b>	Jugador en Entorno Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente del que se desconoce el estado
Motivo	Motivo por el que se desconoce el estado

**Mensaje 5 : Estado Compartido Desconocido**

**M6**

<b>PedirReinicio</b>	
<b>Descripción</b>	Un agente jugador pide que su posición sea reiniciada.
<b>Tipo</b>	REQUEST
<b>Emisor</b>	Jugador en Entorno Exclusivo o Jugador en Entorno Compartido
<b>Receptor</b>	Agente Reinicio
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que pide ser reiniciado
Equipo	Equipo al que pertenece el agente que pide ser reiniciado

**Mensaje 6 : Pedir Reinicio**

**M7**

<b>ReinicioIncorrecto</b>	
<b>Descripción</b>	El agente reinicio informa de que el reinicio no se ha podido producir.
<b>Tipo</b>	REFUSE
<b>Emisor</b>	Agente Reinicio
<b>Receptor</b>	Jugador en Entorno Exclusivo o Jugador en Entorno Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que pide ser reiniciado
Equipo	Equipo al que pertenece el agente que pide ser reiniciado
Motivo	Motivo por el que no pudo ser reiniciado.

**Mensaje 7 : Reinicio Incorrecto**

**M8**

<b>PedirMovimiento</b>	
<b>Descripción</b>	Un agente Jugador solicita ejecutar una acción sobre el entorno Tablero.
<b>Tipo</b>	REQUEST
<b>Emisor</b>	Jugador en Entorno Exclusivo o Jugador en Entorno Compartido
<b>Receptor</b>	Agente Movimiento
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que pide ser movido
Movimiento	Movimiento que quiere realizar
Equipo	Equipo al que pertenece el agente que solicita el movimiento

**Mensaje 8 : Pedir Movimiento**

**M9**

<b>MovimientoCorrecto</b>	
<b>Descripción</b>	El Agente Movimiento confirma que el movimiento se ha realizado con éxito.
<b>Tipo</b>	CONFIRM
<b>Emisor</b>	Agente Movimiento
<b>Receptor</b>	Jugador en Entorno Exclusivo o Jugador en Entorno Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que ha sido movido
Movimiento	Movimiento realizado
Refuerzo	Refuerzo obtenido al efectuar el movimiento

**Mensaje 9 : Movimiento Correcto**

**M10**

<b>MovimientoIncorrecto</b>	
<b>Descripción</b>	El Agente Movimiento rechaza el movimiento pedido.
<b>Tipo</b>	REFUSE
<b>Emisor</b>	Agente Movimiento
<b>Receptor</b>	Jugador en Entorno Exclusivo o Jugador en Entorno Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que no ha podido ser movido
Movimiento	Movimiento que no ha podido llevarse a cabo
Refuerzo	Refuerzo obtenido al no efectuar el movimiento
Motivo	Motivo por el que no se pudo realizar el movimiento

**Mensaje 10 : Movimiento Incorrecto**

M11	
EnviarEstadísticasExclusivas	
<b>Descripción</b>	El agente Jugador en Entorno Exclusivo envía las estadísticas de un episodio de aprendizaje al agente Gestor Estadístico.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Jugador en Entorno Exclusivo
<b>Receptor</b>	Gestor Estadístico
Lista de Argumentos	
Argumento	Descripción
Agente	Agente que ha generado las estadísticas
Estadísticas	Estadísticas del episodio acabado

Mensaje 11 : Enviar Estadísticas Exclusivas

M12	
EnviarEstadísticasCompartidas	
<b>Descripción</b>	El agente Jugador en Entorno Compartido envía las estadísticas de un episodio de aprendizaje colectivo al agente Gestor Estadístico.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Jugador en Entorno Compartido
<b>Receptor</b>	Gestor Estadístico
Lista de Argumentos	
Argumento	Descripción
Agente	Agente que ha generado las estadísticas
Estadísticas	Estadísticas del episodio colectivo acabado

Mensaje 12 : Enviar Estadísticas Compartidas

M13	
ActualizarGráfica	
<b>Descripción</b>	El agente Control del Entrenamiento en Entorno Compartido solicita al Procesador Estadístico que actualice la gráfica en la que se muestran las estadísticas de los procesos de aprendizaje al usuario.
<b>Tipo</b>	REQUEST
<b>Emisor</b>	Control del Entrenamiento en Entorno Compartido
<b>Receptor</b>	Procesador Estadístico
Lista de Argumentos	
Argumento	Descripción
Agente	Agente que solicita actualizar la gráfica.

Mensaje 13 : Actualizar Gráfica

**M14**

<b>EnviarPolíticaExclusiva</b>	
<b>Descripción</b>	Un agente Jugador en Entorno Exclusivo envia al Gestor de Conocimiento Exclusivo la política generada durante su proceso de aprendizaje.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Jugador en Entorno Exclusivo
<b>Receptor</b>	Gestor de Conocimiento Exclusivo
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que ha generado el conocimiento.
Equipo	Equipo al que pertenece
Política	Política de acciones generada

**Mensaje 14 : Enviar Política Exclusiva**

**M15**

<b>EnviarPolíticaCompartida</b>	
<b>Descripción</b>	Un agente Jugador en Entorno Compartido envia al Gestor de Conocimiento Compartido la política generada durante su proceso de aprendizaje.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Jugador en Entorno Compartido
<b>Receptor</b>	Gestor de Conocimiento Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que ha generado el conocimiento.
Equipo	Equipo al que pertenece
Política	Política de acciones generada

**Mensaje 15 : Enviar Política Compartida**

**M16**

<b>PedirMejorMovimientoExclusivo</b>	
<b>Descripción</b>	Un agente Jugador en Entorno Compartido solicita al Gestor de Conocimiento Exclusivo que le informe sobre la mejor acción en un estado concreto según la política de acciones en entorno exclusivo que conoce.
<b>Tipo</b>	REQUEST
<b>Emisor</b>	Jugador en Entorno Compartido
<b>Receptor</b>	Gestor de Conocimiento Exclusivo
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que hace la solicitud.
Equipo	Equipo al que pertenece
Estado	Estado del que quiere conocer la mejor accion

**Mensaje 16 : Pedir Mejor Movimiento Exclusivo**

**M17**

<b>PedirPolíticaCompartida</b>	
<b>Descripción</b>	Un agente Jugador en Entorno Compartido solicita al Gestor de Conocimiento Compartido que le informe sobre la mejor acción en un estado concreto según la política de acciones en entorno exclusivo que conoce.
<b>Tipo</b>	REQUEST
<b>Emisor</b>	Jugador en Entorno Compartido
<b>Receptor</b>	Gestor de Conocimiento Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que hace la solicitud.
Equipo	Equipo al que pertenece

**Mensaje 17 : Pedir Política Compartida**

**M18**

<b>ListoParaEntrenamientoEnEntornoExclusivo</b>	
<b>Descripción</b>	El agente Jugador en Entorno Exclusivo informa a Control del Entrenamiento en Entorno Exclusivo de que se encuentra preparado para empezar el entrenamiento en entorno Exclusivo.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Jugador en Entorno Exclusivo
<b>Receptor</b>	Control del Entrenamiento en Entorno Exclusivo
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que se encuentra preparado.

**Mensaje 18 : Listo Para Entrenamiento En Entorno Exclusivo**

**M19**

<b>EmpezarEntrenamientoEntornoExclusivo</b>	
<b>Descripción</b>	El agente Control del Entrenamiento en Entorno Exclusivo informa a un agente Jugador en Entorno Exclusivo de que puede empezar su entrenamiento.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Control del Entrenamiento en Entorno Exclusivo
<b>Receptor</b>	Jugador en Entorno Exclusivo
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que puede empezar el entrenamiento.

**Mensaje 19 : Empezar Entrenamiento Entorno Exclusivo**

**M20**

<b>EntrenamientoEntornoExclusivoAcabado</b>	
<b>Descripción</b>	El agente Jugador en Entorno Exclusivo informa a Control del Entrenamiento en Entorno Exclusivo de que ha acabado su proceso de aprendizaje.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Jugador en Entorno Exclusivo
<b>Receptor</b>	Control del Entrenamiento en Entorno Exclusivo
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que ha acabado el entrenamiento

**Mensaje 20 : Entrenamiento Entorno Exclusivo Acabado**



**M21**

<b>ListoParaEpisodioEntrenamientoEnEntornoCompartido</b>	
<b>Descripción</b>	El agente Jugador en Entorno Compartido informa a Control del Entrenamiento en Entorno Compartido de que se encuentra preparado para empezar un episodio del entrenamiento en entorno Compartido.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Jugador en Entorno Compartido
<b>Receptor</b>	Control del Entrenamiento en Entorno Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que se encuentra preparado para el episodio.

**Mensaje 21 : Listo Para Episodio Entrenamiento En Entorno Compartido**

**M22**

<b>EmpezarEpisodioEntrenamientoEnEntornoCompartido</b>	
<b>Descripción</b>	El agente Control del Entrenamiento en Entorno Compartido informa a un agente Jugador en Entorno Compartido de que puede empezar un episodio de su entrenamiento.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Control del Entrenamiento en Entorno Compartido
<b>Receptor</b>	Jugador en Entorno Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que puede empezar el episodio de entrenamiento.

**Mensaje 22 : Empezar Episodio Entrenamiento Entorno Compartido**

**M23**

<b>AcabadoEpisodioEntrenamientoEnEntornoCompartido</b>	
<b>Descripción</b>	El agente Jugador en Entorno Compartido informa a Control del Entrenamiento en Entorno Compartido de que ha acabado un episodio de su proceso de aprendizaje.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Jugador en Entorno Compartido
<b>Receptor</b>	Control del Entrenamiento en Entorno Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que ha acabado su episodio de entrenamiento.

**Mensaje 23 : Acabado Episodio Entrenamiento En Entorno Compartido**

**M24**

<b>ListoParaCicloEntrenamientoEnEntornoCompartido</b>	
<b>Descripción</b>	El agente Jugador en Entorno Compartido informa a Control del Entrenamiento en Entorno Compartido de que se encuentra preparado para empezar un nuevo ciclo de entrenamiento.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Jugador en Entorno Compartido
<b>Receptor</b>	Control del Entrenamiento en Entorno Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que se encuentra preparado para el ciclo.

**Mensaje 24 : Listo Para Ciclo Entrenamiento En Entorno Compartido**

**M25**

<b>EmpezarCicloEntrenamientoEnEntornoCompartido</b>	
<b>Descripción</b>	El agente Control informa a un agente Jugador Colectivo de que puede empezar su ciclo de entrenamiento colectivo.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Control del Entrenamiento en Entorno Compartido
<b>Receptor</b>	Jugador en Entorno Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que puede empezar el ciclo de entrenamiento.

**Mensaje 25 : Empezar Ciclo Entrenamiento En Entorno Compartido**

**M26**

<b>AcabadoCicloEntrenamientoEnEntornoCompartido</b>	
<b>Descripción</b>	El agente Jugador en Entorno Compartido informa a Control del Entrenamiento en Entorno Compartido de que ha acabado su ciclo de entrenamiento.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Jugador en Entorno Compartido
<b>Receptor</b>	Control del Entrenamiento en Entorno Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que ha acabado su ciclo de entrenamiento colectivo.

**Mensaje 26 : Acabado Ciclo Entrenamiento En Entorno Compartido**

**M27**

<b>EntrenamientoEnEntornoCompartidoAcabado</b>	
<b>Descripción</b>	El agente Jugador en Entorno Compartido informa a Control del Entrenamiento en Entorno Compartido de que su entrenamiento ha finalizado.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Jugador en Entorno Compartido
<b>Receptor</b>	Control del Entrenamiento en Entorno Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que ha acabado su entrenamiento.

**Mensaje 27 : Entrenamiento En Entorno Compartido Acabado**

**M28**

<b>PedirRegistroEnEntorno</b>	
<b>Descripción</b>	El agente jugador pide su registro en el entorno Tablero
<b>Tipo</b>	REQUEST
<b>Emisor</b>	Jugador en Entorno Compartido o Jugador en Entorno Exclusivo
<b>Receptor</b>	Control del Entrenamiento en Entorno Compartido o Control del Entrenamiento en Entorno Exclusivo
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que pide su registro en el entorno Tablero

**Mensaje 28 : Pedir Registro En Entorno**

**M29**

<b>AgenteRegistrado</b>	
<b>Descripción</b>	El agente control confirma que el agente jugador ha sido registrado en el entorno Tablero
<b>Tipo</b>	CONFIRM
<b>Emisor</b>	Control del Entrenamiento en Entorno Compartido o Control del Entrenamiento en Entorno Exclusivo
<b>Receptor</b>	Jugador en Entorno Compartido o Jugador en Entorno Exclusivo
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que ha sido registrado

**Mensaje 29 : Agente Registrado**

**M30**

<b>AgenteNoRegistrado</b>	
<b>Descripción</b>	El agente control rechaza la petición de registro del agente jugador.
<b>Tipo</b>	REFUSE
<b>Emisor</b>	Jugador en Entorno Compartido o Jugador en Entorno Exclusivo
<b>Receptor</b>	Control del Entrenamiento en Entorno Compartido o Control del Entrenamiento en Entorno Exclusivo
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que ha sido rechazado
Motivo	Motivo por el que el agente no ha sido registrado

**Mensaje 30 : Agente No Registrado**

**M31**

<b>SolicitarUsoAgente</b>	
<b>Descripción</b>	El agente emisor pide permiso al Servidor para usar sus servicios durante la ejecución.
<b>Tipo</b>	SUBSCRIBE
<b>Emisor</b>	Jugador en Entorno Compartido o Jugador en Entorno Exclusivo
<b>Receptor</b>	Servidor
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que pide autorización

**Mensaje 31 : Solicitar Uso Agente**

**M32**

<b>AgenteAutorizado</b>	
<b>Descripción</b>	El agente Servidor confirma que el agente receptor puede usar sus servicios.
<b>Tipo</b>	CONFIRM
<b>Emisor</b>	Servidor
<b>Receptor</b>	Jugador en Entorno Compartido o Jugador en Entorno Exclusivo
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que ha sido autorizado a usar los servicios del servidor.
Agente	Agente que actuará como servidor del receptor del mensaje

**Mensaje 32 : Agente Autorizado**

**M33**

<b>AgenteNoAutorizado</b>	
<b>Descripción</b>	El agente Servidor rechaza la petición de servicios que efectuó el agente receptor.
<b>Tipo</b>	REFUSE
<b>Emisor</b>	Servidor
<b>Receptor</b>	Jugador en Entorno Compartido o Jugador en Entorno Exclusivo
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que no ha sido autorizado a usar los servicios del Servidor.
Agente	Agente que no actuará como servidor del receptor del mensaje
Motivo	Motivo por el que el agente no ha sido autorizado

**Mensaje 33 : Agente No Autorizado**

**M34**

<b>FinalizarEjecucion</b>	
<b>Descripción</b>	El agente Control solicita al Agente que finalice su ejecución.
<b>Tipo</b>	REQUEST
<b>Emisor</b>	Control del Entrenamiento en Entorno Compartido o Control del Entrenamiento en Entorno Exclusivo
<b>Receptor</b>	Finalizable
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente al que se le solicita que acabe su ejecución
Agente	Agente que solicita que la ejecución cese
Motivo	Motivo por el que ha de finalizar la ejecución.

**Mensaje 34 : Finalizar Ejecución**

**M35**

<b>EjecucionNoFinalizada</b>	
<b>Descripción</b>	El agente rechaza la petición de finalización.
<b>Tipo</b>	REFUSE
<b>Emisor</b>	Finalizable
<b>Receptor</b>	Control del Entrenamiento en Entorno Compartido o Control del Entrenamiento en Entorno Exclusivo
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que va a no finalizar su ejecución.
Agente	Agente que ordenó el cese de la ejecución.
Motivo	Motivo por el que no se va a finalizar la ejecución

**Mensaje 35 : Ejecucion No Finalizada**

**M36**

<b>MejorMovimientoExclusivo</b>	
<b>Descripción</b>	Informa del mejor movimiento en entorno exclusivo según la política de acciones almacenada.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Gestor de Conocimiento en Entorno Exclusivo
<b>Receptor</b>	Agente Jugador en Entorno Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que pidió el conocimiento.
Movimiento	Mejor movimiento conocido.
Estado	Estado para el que es el mejor movimiento conocido.
Equipo	Equipo para el que es el mejor movimiento.

**Mensaje 36 : Mejor Movimiento Exclusivo**

**M37**

<b>MejorPolíticaCompartido</b>	
<b>Descripción</b>	Informa del mejor movimiento en entorno exclusivo según la política de acciones almacenada.
<b>Tipo</b>	INFORM
<b>Emisor</b>	Gestor de Conocimiento en Entorno Compartido
<b>Receptor</b>	Agente Jugador en Entorno Compartido
<b>Lista de Argumentos</b>	
<b>Argumento</b>	<b>Descripción</b>
Agente	Agente que pidió el conocimiento.
Movimiento[]	Mejores movimientos conocidos.
Estado[]	Estados para el que es el mejor movimiento conocido.
Equipo	Equipo para el que es la mejor política.

**Mensaje 37 : Mejor Movimiento Compartido**



## Anexo VII: Ontología

Una ontología es una descripción formal de los conceptos y relaciones que pueden existir en una determinada comunidad de agentes. Su objetivo es compartir conocimiento utilizando un mismo vocabulario de forma coherente y consistente. La ontología del sistema es una garantía de consistencia pero no de completitud con respecto a las preguntas y afirmaciones que se pueden efectuar utilizando el vocabulario definido en una ontología

El objetivo de la ontología es facilitar la comunicación entre agentes de tal forma que dos entidades diferentes se entiendan entre sí y asegurarse de que hablan dentro del mismo contexto.

### VII.1. Conceptos

Los conceptos en una ontología definen partículas simples de información y conocimiento que un agente puede transmitir.

			C1
Agente			
Descripción		Describe un agente jugador dentro del sistema	
Tipo		Concepto	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Nombre	String	Nombre local del jugador.	

Concepto 1 : Agente

C2		
EstadoExclusivo		
Descripción	Describe el estado posición de un agente Jugador en Entorno Exclusivo en el entorno Tablero.	
Tipo	Concepto	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Fila	Int	Fila en la que se encuentra el agente
Columna	int	Columna en la que se encuentra el agente

Concepto 2 : Estado Exclusivo

			C3
Equipo			
Descripción		Describe el equipo al que puede pertenecer un jugador	
Tipo		Concepto	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Identificador	Int	Identificador del equipo	

Concepto 3 : Equipo

**C4**

EstadoCompartido		
Descripción	Describe el estado posición de un agente Jugador en Entorno Compartido en el entorno Tablero.	
Tipo	Concepto	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Coordenada	Coordenada	Coordenadas en las que se encuentra el agente.
Oeste	Int	Situación de la casilla al oeste del jugador.
Este	Int	Situación de la casilla al este del jugador.
Norte	Int	Situación de la casilla al norte del jugador.
Sur	Int	Situación de la casilla al sur del jugador.

**Concepto 4 : Estado Compartido**

**C5**

EstadísticasCicloEntornoCompartido		
Descripción		Describen las estadísticas que un Jugador en Entorno Compartido ha generado en un episodio de aprendizaje
Tipo		Concepto
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente que ha generado las estadísticas.
Equipo	Equipo	Equipo al que pertenece el agente.
Objetivo	Boolean	Indica si la meta ha sido alcanzada en este ciclo (true) o si el ciclo ha acabado por llegar al límite de episodios (false).
Aleatorios	Int	Indica el número de movimientos aleatorios tomados durante todo el episodio.
No aleatorios	Int	Indica el número de movimientos tomados siguiendo políticas no aleatorias.
Exclusivas	Int	Indica el número de movimientos tomados siguiendo la política generada por el aprendizaje en entorno exclusivo.
Compartidas	Int	Indica el número de movimientos tomados siguiendo la política generada por el aprendizaje en entorno compartido.
Total	Int	Indica el número de movimientos efectuados durante el episodio.
No válidos	Int	Indica el número de movimientos no válidos efectuados, es decir, el número de choques contra obstáculos.
Ciclo	Int	Indica el número de episodio al que pertenecen las estadísticas.

**Concepto 5 : Estadísticas Ciclo Entorno Compartido**



**C6**

EstadísticasCicloEntornoExclusivo		
Descripción	Describen las estadísticas que un Jugador en Entorno Exclusivo ha generado en un episodio de aprendizaje	
Tipo	Concepto	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente que ha generado las estadísticas.
Objetivo	Boolean	Indica si la meta ha sido alcanzada en este ciclo (true) o si el ciclo ha acabado por llegar al límite de episodios (false).
Aleatorios	Int	Indica el número de movimientos aleatorios tomados durante todo el episodio.
No aleatorios	Int	Indica el número de movimientos no aleatorios tomados durante todo el episodio.
Total	Int	Indica el número de movimientos efectuados durante el episodio.
No válidos	Int	Indica el número de movimientos no válidos efectuados, es decir, el número de choques contra obstáculos.
Ciclo	Int	Indica el número de episodio al que pertenecen las estadísticas.

Concepto 6 : Estadísticas Ciclo Entorno Exclusivo

**C7**

Motivo		
Descripción	Describe el motivo por el que una decisión fue tomada.	
Tipo	Concepto	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Motivo	String	Descripción del motivo.

Concepto 7 : Motivo

**C8**

Movimiento		
Descripción	Describe un posible movimiento en el entorno	
Tipo	Concepto	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Coordenada	Coordenada	Coordenadas que describen el desplazamiento necesario desde la posición actual del agente.

Concepto 8 : Movimiento

**C9**

Refuerzo		
Descripción	Describe el refuerzo obtenido.	
Tipo	Concepto	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Valor	Int	Valor del refuerzo que se comunica.

Concepto 9 : Refuerzo

			C10
ValorPolítica			
Descripción		Encapsula el valor asociado a una política para un estado.	
Tipo		Concepto	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Valor	Float	Valor de la tabla Q obtenido	
Movimiento	Movimiento	Mejor movimiento conocido para el estado.	

Concepto 10 : Valor Q

			C11
TuplaQExclusiva			
Descripción		Encapsulta una tupla de la función Q.	
Tipo		Concepto	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Estado	Estado Exslusivo	Estado que describe la tupla	
Movimiento	Movimiento	Movimiento asociado a la tupla.	

Concepto 11 : Tupla Q Exclusiva

			C12
TuplaQCompartida			
Descripción		Encapsulta una tupla de la función Q.	
Tipo		Concepto	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Estado	Estado Compartido	Estado que describe la tupla	
Movimiento	Movimiento	Movimiento asociado a la tupla.	

Concepto 12 : Tupla Q Compartida

## VII.2. Acciones

Las acciones dentro de una ontología definen las interacciones posibles entre agentes. Siempre contienen el agente que inicia la interacción y pueden más argumentos si fueran necesarios.

			A1
EmpezarEntrenamientoEntornoExclusivo			
Descripción		El agente Control del Entrenamiento en Entorno Exclusivo pide a un agente Jugador en Entorno Exclusivo que empiece con su entrenamiento.	
Tipo		Acción	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Agente	Agente	Agente que puede empezar su entrenamiento.	

Acción 1 : Empezar Entrenamiento Entorno Exclusivo

			A2
EnviarEstadísticasExclusivas			
Descripción		Un agente Jugador en Entorno Exclusivo envía las estadísticas de un ciclo de aprendizaje.	
Tipo		Acción	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Estadísticas	Estadísticas Ciclo Exclusivo	Estadísticas que describen el ciclo individual de aprendizaje realizado por el agente Jugador Individual	

Acción 2 : Enviar Estadísticas Exclusivas

			A3
FinalizarEjecucion			
Descripción		El agente Control pide a un agente que finalice su ejecución.	
Tipo		Acción	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Agente	Agente	Agente que el agente Control quiere que finalice su ejecución.	
Ordenante	Agente	Agente que ha ordenado que finalice la ejecución.	
Motivo	Motivo	Motivo por el que ha de finalizar la ejecución.	

Acción 3 : Finalizar Ejecución

			A4
EntrenamientoExclusivoAcabado			
Descripción		Un agente Jugador en Entorno Exclusivo informa de que ha acabado su entrenamiento.	
Tipo		Acción	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Agente	Agente	Agente que informa de que su entrenamiento ha acabado.	

Acción 4 : Entrenamiento Exclusivo Acabado

			A5
ListoParaEntrenamientoEntornoExclusivo			
Descripción		Un agente Jugador en Entorno Exclusivo informa de que se encuentra listo para empezar el entrenamiento.	
Tipo		Acción	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Agente	Agente	Agente Jugador en Entorno Exclusivo listo para empezar el entrenamiento.	

Acción 5 : Listo Para Entrenamiento Entorno Exclusivo

			A6
PedirEstado			
Descripción		Un agente Jugador pide ser informado acerca de su estado.	
Tipo		Acción	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Agente	Agente	Agente que pide ser informado acerca de su estado.	

Acción 6 : Pedir Estado

A7		
PedirMovimiento		
Descripción	Un agente Jugador pide un movimiento	
Tipo	Acción	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente que pide ser movido.
Movimiento	Movimiento	Movimiento que pide.
Equipo	Equipo	Equipo al que pertenece.

Acción 7 : Pedir Movimiento

			A8
PedirRegistroEnTablero			
Descripción		Un agente pide ser registrado en el entorno Tablero	
Tipo		Acción	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Agente	Agente	Agente que pide ser registrado en el entorno Tablero.	

Acción 8 : Pedir Registro En Tablero

			A9
Reiniciar			
Descripción		Un agente Jugador pide que su posición sea reiniciada	
Tipo		Acción	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Agente	Agente	Agente que pide que su posición sea reiniciada.	
Equipo	Equipo	Equipo al que pertenece el agente.	

Acción 9 : Reiniciar

			A10
SolicitarUsoAgente			
Descripción		Un agente pide permiso para usar los servicios de otro agente.	
Tipo		Acción	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Agente	Agente	Agente que pide usar los servicios del otro agente.	

Acción 10 : Solicitar Uso Agente

A11		
ListoParaCicloEntrenamientoEnEntornoCompartido		
Descripción	Un agente Jugador en Entorno Compartido informa de que se encuentra preparado para empezar un ciclo de entrenamiento.	
Tipo	Acción	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente Jugador en Entorno Compartido que se encuentra listo para empezar el ciclo de entrenamiento.

Acción 11 : Listo Para Ciclo Entrenamiento En Entorno Compartido

**A12**

EmpezarCicloEntrenamientoEnEntornoCompartido		
Descripción		El agente Control del Entrenamiento en Entorno Compartido informa a un agente Jugador en Entorno Compartido de que puede empezar su ciclo de entrenamiento.
Tipo		Acción
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente Jugador en Entorno Compartido que puede empezar su ciclo de entrenamiento.

**Acción 12 : Empezar Ciclo Entrenamiento En Entorno Compartido**

**A13**

AcabadoCicloEntrenamientoEnEntornoCompartido		
Descripción		Un agente Jugador en Entorno Compartido informa de que ha acabado su ciclo de entrenamiento.
Tipo		Acción
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente que ha acabado su ciclo de entrenamiento.

**Acción 13 : Acabado Ciclo Entrenamiento En Entorno Compartido**

**A14**

ListoParaEpisodioEntrenamientoEnEntornoCompartido		
Descripción		Un agente Jugador en Entorno Compartido informa de que se encuentra preparado para empezar un episodio de entrenamiento.
Tipo		Acción
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente que se encuentra listo para empezar el episodio de entrenamiento.

**Acción 14 : Listo Para Episodio Entrenamiento En Entorno Compartido**

**A15**

EmpezarEpisodioEntrenamientoEnEntornoCompartido		
Descripción		El agente Control del Entrenamiento en Entorno Compartido informa a un agente Jugador en Entorno Compartido de que puede empezar su episodio de entrenamiento.
Tipo		Acción
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente Jugador en Entorno Compartido que puede empezar su episodio de entrenamiento.

**Acción 15 : Empezar Episodio Entrenamiento En Entorno Compartido**

**A16**

AcabadoEpisodioEntrenamientoEnEntornoCompartido		
Descripción		Un agente Jugador en Entorno Compartido informa de que ha acabado su episodio de entrenamiento en Entorno Compartido.
Tipo		Acción
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente que ha acabado su episodio de entrenamiento.

**Acción 16 : Acabado Episodio Entrenamiento En Entorno Compartido**

**A17**

EnviarEstadísticasCompartidas		
Descripción	Un agente Jugador en Entorno Compartido envía las estadísticas de un episodio de aprendizaje	
Tipo	Acción	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Estadísticas	Estadísticas Ciclo Compartido	Estadísticas que describen el ciclo colectivo de aprendizaje realizado por el agente Jugador en Entorno Compartido

**Acción 17 : Enviar Estadísticas Compartidas**

**A18**

EntrenamientoCompartidoAcabado		
Descripción		Un agente Jugador en Entorno Compartido informa de que ha acabado su entrenamiento colectivo.
Tipo		Acción
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente que informa de que su entrenamiento ha acabado.

**Acción 18 : Entrenamiento Compartido Acabado**

### VII.3. Predicados

Los predicados dentro de una ontología son expresiones que relacionan conceptos.

**P1**

AgenteAutorizado		
Descripción	Describe la autorización dada por un agente a otro para usar sus servicios.	
Tipo	Predicado	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Cliente	Agente	Agente que ha sido autorizado
Servidor	Agente	Agente que dará servicio.

**Predicado 1 : Agente Autorizado**

**P2**

AgenteNoAutorizado		
Descripción	Describe la autorización rechazada por un agente a otro para usar sus servicios.	
Tipo	Predicado	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente que no ha sido autorizado
Servidor	Agente	Agente que no dará servicio.
Motivo	Motivo	Motivo por el que no ha sido autorizado

**Predicado 2 : Agente No Autorizado**

**P3**

AgenteRegistrado		
Descripción	Describe un agente que ha sido registrado en el entorno.	
Tipo	Predicado	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente que ha sido registrado.
Equipo	Equipo	Equipo en el que ha sido incluido-

**Predicado 3 : Agente Registrado**

**P4**

AgenteNoRegistrado		
Descripción	Describe un agente que no ha sido registrado en el entorno.	
Tipo	Predicado	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente que no ha sido registrado.
Motivo	Motivo	Motivo por el que no ha sido registrado

**Predicado 4 : Agente No Registrado**

**P5**

AgentePerteneceEquipo		
Descripción	Describe a que equipo pertenece un agente dado.	
Tipo	Predicado	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente que pertenece al equipo
Equipo	Equipo	Equipo al que pertenece

**Predicado 5 : Agente Pertenece Equipo**

**P6**

EjecucionFinalizada		
Descripción	Indica que un agente ha acabado su ejecución.	
Tipo	Predicado	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente que ha acabado su ejecución.
Agente	Ordenante	Agente que ha ordenado la finalización.

**Predicado 6 : Ejecucion Finalizada**

**P7**

EjecucionNoFinalizada		
Descripción	Indica que un agente no ha acabado su ejecución.	
Tipo	Predicado	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente que no ha acabado su ejecución.
Agente	Ordenante	Agente que ha ordenado la finalización.
Motivo	Motivo	Motivo por el que la ejecución no ha cesado.

**Predicado 7 : Ejecucion No Finalizada**

**P8**

EstadoAgenteCompartido		
Descripción	Asocia un estado a un agente Jugador en Entorno Compartido	
Tipo	Predicado	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente del que se describe su entorno
Estado	Estado Compartido	Estado del agente.

**Predicado 8 : Estado Agente Compartido**

**P9**

EstadoAgenteExclusivo		
Descripción	Asocia un estado a un agente Jugador en Entorno Exclusivo	
Tipo	Predicado	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente del que se describe su entorno
Estado	Estado Exclusivo	Estado del agente.

**Predicado 9 : Estado Agente Exclusivo**

**P10**

EstadoDesconocido		
Descripción	Indica que el estado de un agente es desconocido.	
Tipo	Predicado	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente del que se desconoce el estado.
Motivo	Motivo	Motivo por el que se desconoce.

**Predicado 10 : Estado Desconocido**



Inicio			P11
Descripción		Describe un estado como estado de inicio.	
Tipo		Concepto	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Coordenada	Coordenada	Estado que será de inicio.	
Equipo	Equipo	Equipo de jugadores del que será inicio es estado.	

Predicado 11 : Inicio

MejorMovimiento			P12
Descripción		Describe el mejor movimiento conocido para un estado dado.	
Tipo		Predicado	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Coordenada	Coordenada	Estado para la que se conoce el movimiento	
Movimiento	Movimiento	Mejor movimiento conocido para el estado dado.	
ValorQ	Valor	Valor en la matriz Q para el estado y movimiento dado	

Predicado 12 : Mejor Movimiento

			P13
Meta			
Descripción		Describe un estado como estado final.	
Tipo		Concepto	
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Coordenada	Coordenada	Estado que será final.	
Equipo	Equipo	Equipo del que la casilla será final.	

Predicado 13 : Meta

MovimientoCorrecto			P14
Descripción	Indica que el movimiento fue efectuado correctamente.		
Tipo	Predicado		
Lista de Parámetros			
Parámetro	Tipo	Descripción	
Agente	Agente	Agente que ha realizado el movimiento.	
Movimiento	Movimiento	Movimiento que se ha realizado.	
Refuerzo	Refuerzo	Refuerzo obtenido como consecuencia del movimiento.	

Predicado 14 : Movimiento Correcto

**P15**

MovimientoIncorrecto		
Descripción	Indica que el movimiento no se ha podido realizar.	
Tipo	Predicado	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente que no se ha podido mover.
Movimiento	Movimiento	Movimiento que no se ha realizado.
Refuerzo	Refuerzo	Refuerzo obtenido como consecuencia del movimiento.
Motivo	Motivo	Motivo por le que no se pudo realizar el movimiento

**Predicado 15 : Movimiento Incorrecto**

**P16**

ReiniciolIncorrecto		
Descripción	Describe porqué una acción de reinicio ha sido incorrecta.	
Tipo	Predicado	
Lista de Parámetros		
Parámetro	Tipo	Descripción
Agente	Agente	Agente que ordenó el reinicio.
Equipo	Equipo	Equipo al que pertenece el agente.
Motivo	Motivo	Motivo por el que el reinicio es incorrecto.

**Predicado 16 : Reinicio Incorrecto**

## ANEXO VIII: Notación de las normas de la organización

Para las normas descritas a continuación se sigue la siguiente notación, propuesta en [7]. La notación de las **normas del servicio** es la siguiente.

- $R \rightarrow Q$ : El rol Q sólo puede ser ejecutado por una entidad si antes ha ejecutado el rol R.
- $R^n \rightarrow Q$ : El rol Q sólo puede ser ejecutado por una entidad si antes ha ejecutado el rol R n veces.
- $\neg(R \mid Q)$ : Los roles Q y R no pueden ser ejecutados al mismo tiempo
- $\neg((R|0) \mid (Q|0))$ : El rol Q no puede ser ejecutado si no se ejecuta el rol R y viceversa. Es decir, un agente debe ejecutar los dos roles al mismo tiempo.
- $R^{1..N}$ : El rol R puede ser ejecutado al menos una vez y nunca más de N veces.

La notación de las **normas de las interacciones** es la siguiente:

- $P^1$ : La interacción P sólo debe ser ejecutada una vez.
- $P(R)^1$ : La interacción P sólo debe ser ejecutada una vez por el rol R.
- $P \rightarrow Q$ : La ejecución de la interacción P debe preceder a la ejecución de la interacción Q.
- $P(R) \rightarrow Q(R)$ : La interacción P debe ser ejecutada por el rol R antes de que R pueda ejecutar la interacción Q.
- $P(R) \rightarrow [Q(R)] \rightarrow T(R)$ : Entre la ejecución de la interacción P y T por parte del rol R puede ejecutarse la interacción Q.



## ANEXO IX: Gestión y planificación del proyecto

En este anexo se detalla tanto la gestión de los recursos necesarios para la realización del proyecto como la planificación del mismo.

### IX.1 Recursos materiales

Para la realización de este proyecto han sido necesarios múltiples recursos materiales, los cuales se detallan a continuación junto con su coste en la tabla 125.

Recurso	Cantidad	Precio unitario (€)	Precio total (€)
Ordenador	1	1230	1230
Impresora	1	130	130
Cartucho de tinta negra	3	32.45	97.35
Cartucho de tinta color	1	45.25	45.25
Tarrina de 50 CD's	1	18.75	18.75
Memoria USB	1	15.30	15.30
Paquete de folios	3	3.55	10.65
<b>Total</b>			<b>1547.3€</b>

Tabla 125: Recursos materiales empleados

### IX.2 Planificación

Para el desarrollo del sistema han sido necesario un Jefe de Proyecto (J), un Analista (A) y un programador (P) con jornadas de trabajo de 5 horas. La tabla 126 concreta las fases en las que son necesarios cada uno de los trabajadores y su coste asociado. Además en la ilustración 63 se incluye el diagrama de Gantt que se utilizó a la hora de planificar las diferentes fases del proyecto.

Tarea	Inicio	Fin	Horas de trabajo	Asignado a			Coste (€)
				J	A	P	
Visión general del sistema	01/10/2009	10/11/2009	145	X			5075
Análisis del sistema	11/11/2009	25/01/2010	270		X		8910
Diseño arquitectónico	26/01/2010	01/03/2010	125		X		4125
Diseño detallado	02/03/2010	30/04/2010	220		X		7260
Implementación	03/05/2010	29/06/2010	210			X	6090
Pruebas de funcionamiento	01/09/2010	17/09/2010	65			X	1885
Experimentación	24/09/2010	30/09/2010	25		X		825
Evaluación y conclusiones	01/10/2010	21/10/2010	75	X			2625
<b>Total</b>				<b>220</b>	<b>640</b>	<b>275</b>	<b>36795€</b>

Tabla 126: Planificación temporal del proyecto y sus costes

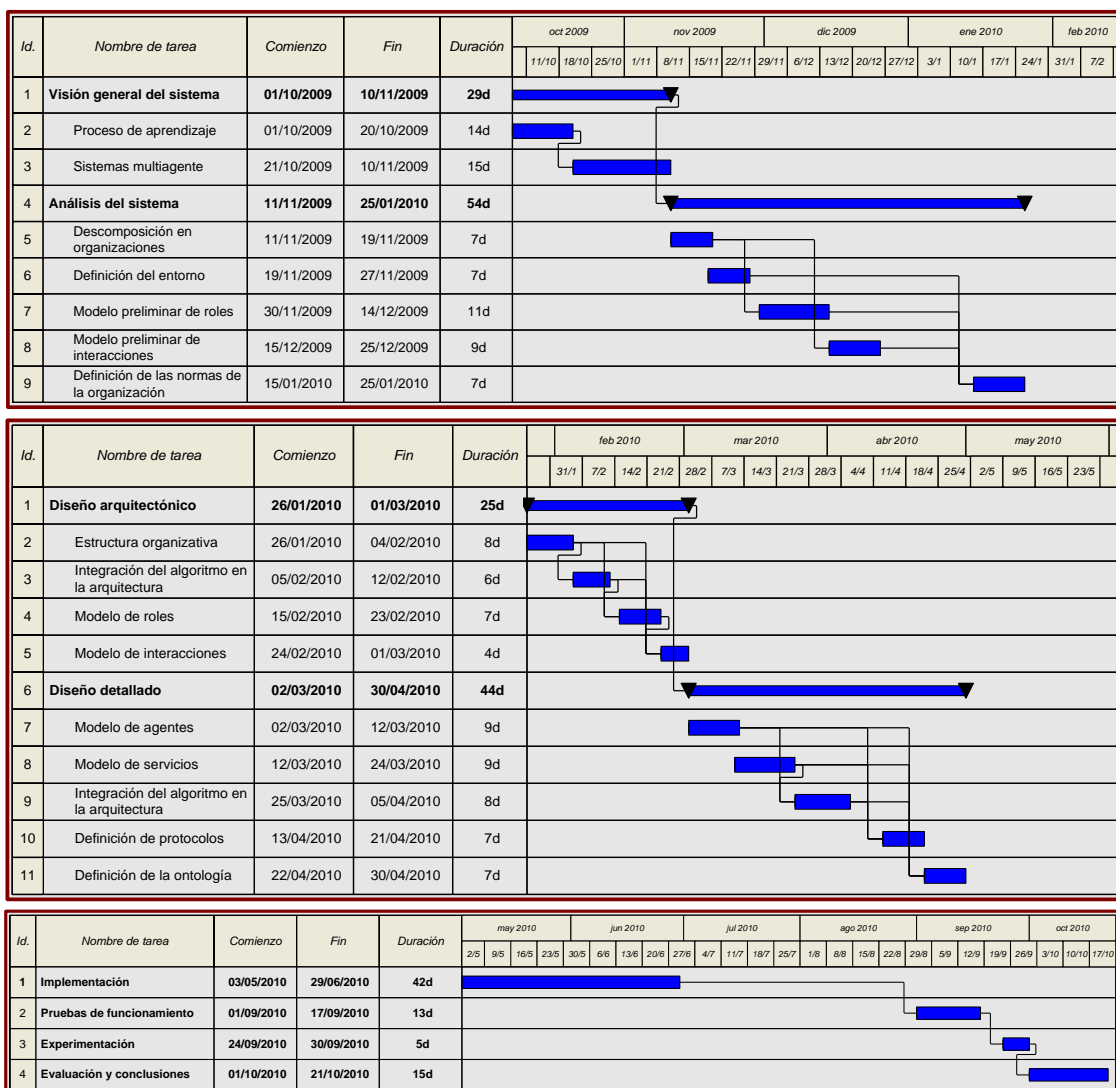


Ilustración 63: Planificación del proyecto en diagramas de Gantt

### IX.3 Recursos humanos

Para la realización de este proyecto han sido necesarios tres personas entre jefe de proyecto, analista y programador. El coste asociado a su puesto de trabajo se ha calculado teniendo en cuenta el BOE 063 de 13/03/2008.

Recurso	Cantidad	Coste bruto por hora (€)	Horas de trabajo	Coste bruto total (€)
Jefe de proyecto	1	35	220	7700
Analista	1	33	640	21120
Programador	1	29	275	7975
<b>Total</b>			<b>1135</b>	<b>36795€</b>

## **IX.4 Totales**

El desarrollo del proyecto emplea 1135 horas de trabajo repartidas entre un Jefe de Proyecto (220 horas), un Analista (640 horas) y un Programador (275 horas).

El desarrollo del proyecto necesita 38342.3€ repartidos entre 1547.3€ de gastos materiales y 36795€ en recursos humanos.